



LAB 1

CONSTRUCT A SIMPLE NETWORK

Name: Trương Đăng Trúc Lâm

ID: B2111933

Group: M04

*Submission: an **ID_NAME_Lab01.pdf** file describes clearly how did you solve the problem*

Exercise 0: change the directory to your home directory

Answer: \$cd

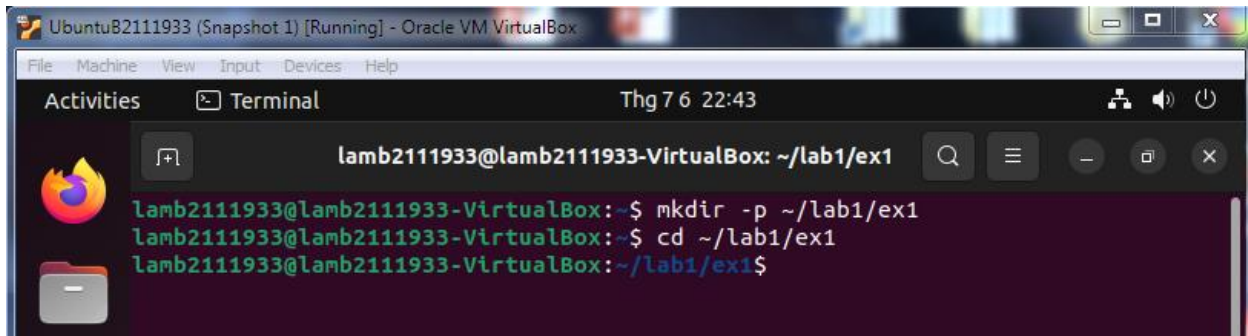


Exercise 1: Construct a simple network with two hosts connected to the same collision domain**Answer:**

Create the directory `~/lab1/ex1`:

```
$ mkdir -p ~/lab1/ex1
```

```
$ cd ~/lab1/ex1
```

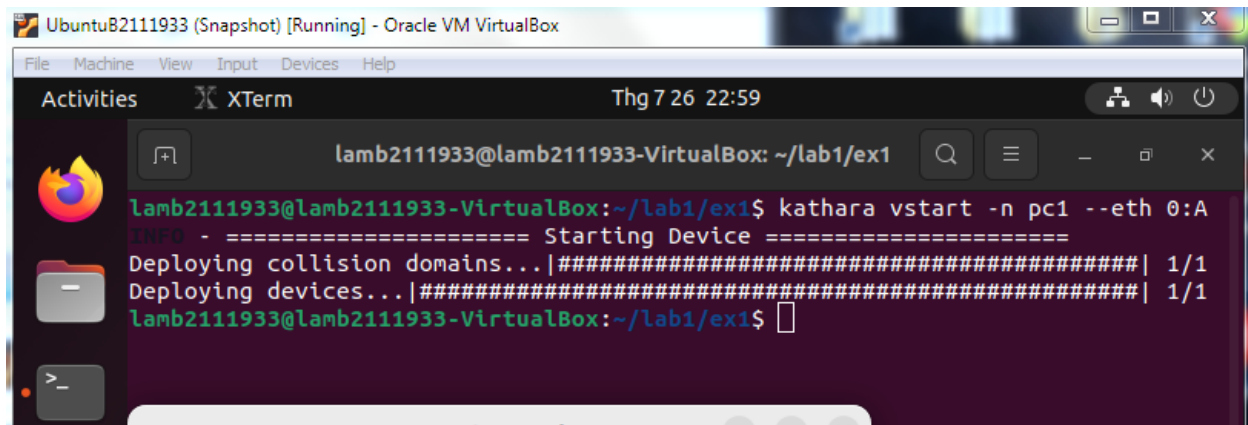


The screenshot shows a terminal window titled "UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex1
lamb2111933@lamb2111933-VirtualBox:~$ mkdir -p ~/lab1/ex1
lamb2111933@lamb2111933-VirtualBox:~$ cd ~/lab1/ex1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex1$
```

Start a new device called **pc1** and connected to the virtual collision domain **A**:

```
$ kathara vstart -n pc1 --eth 0:A
```

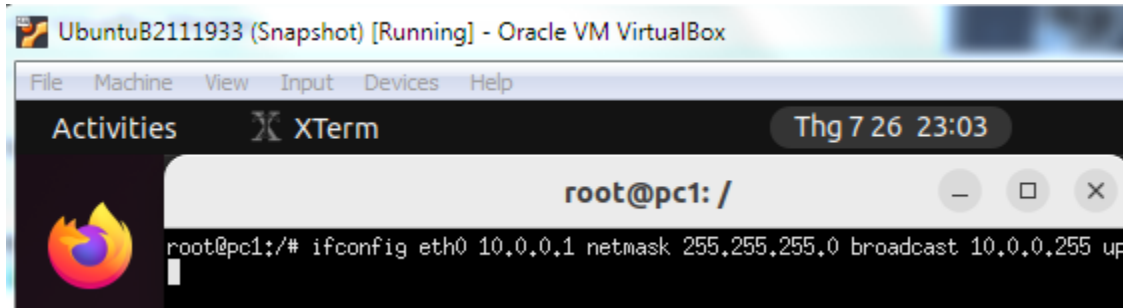


The screenshot shows a terminal window titled "UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex1$ kathara vstart -n pc1 --eth 0:A
INFO - ===== Starting Device =====
Deploying collision domains...|#####| 1/1
Deploying devices...|#####| 1/1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex1$
```

Configure network interfaces for **pc1**:

```
# ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up
```



The screenshot shows a terminal window titled "UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

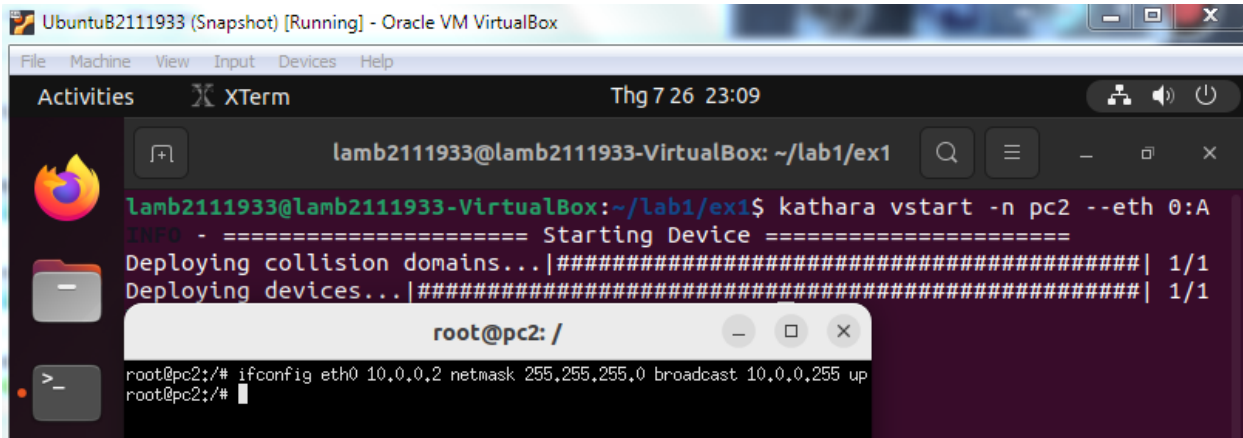
```
root@pc1: /
root@pc1:/# ifconfig eth0 10.0.0.1 netmask 255,255,255,0 broadcast 10.0.0.255 up
```

Start another device called **pc2** and connected to the virtual collision domain **A**

```
$ kathara vstart -n pc2 --eth 0:A
```

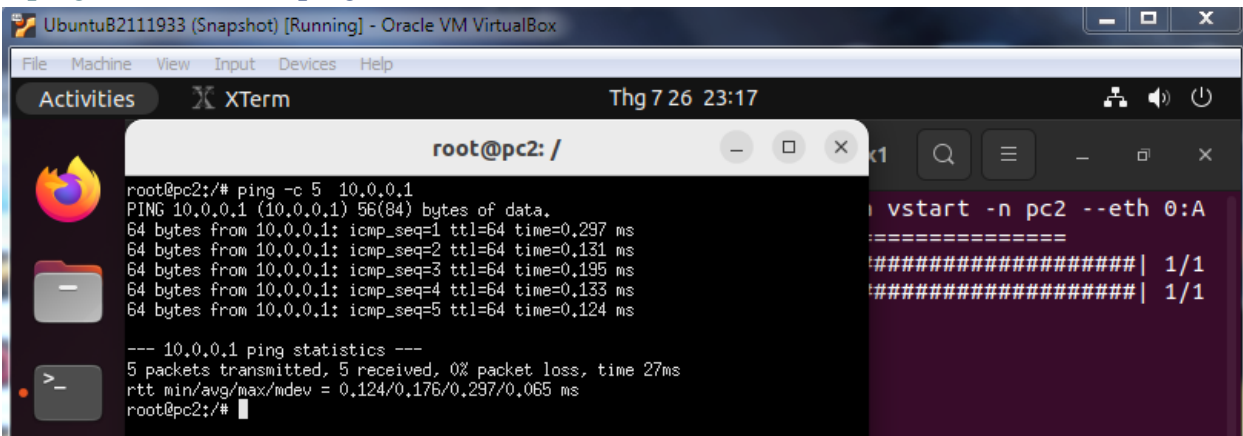
Then configure it's network interfaces:

```
# ifconfig eth0 10.0.0.2 netmask 255.255.255.0 broadcast 10.0.0.255 up
```

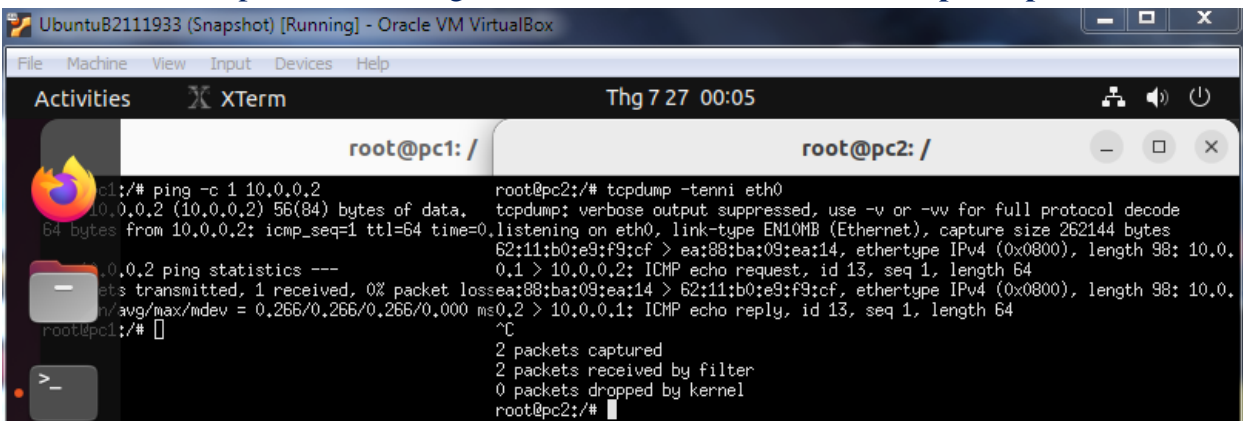


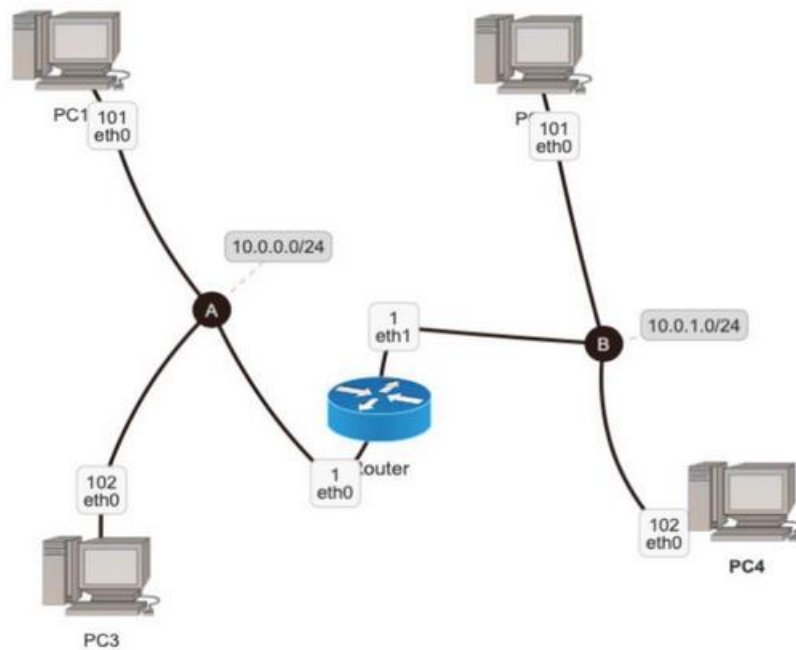
Test connectivity from **pc2** to **pc1**:

```
# ping -c 5 10.0.0.1 (ping 5 times)
```



We can sniff the packets exchanged on collision domain **A** with “**tcpdump**” command:



Exercise 2: Construct the network**Answer:**

Prepare a lab for **Exercise 2**:

```
$ mkdir pc1 pc2 pc3 pc4 router1 shared
```

```
$ gedit lab.conf
```

```
$ gedit pc1.startup
```

```
$ gedit pc2.startup
```

```
$ gedit pc3.startup
```

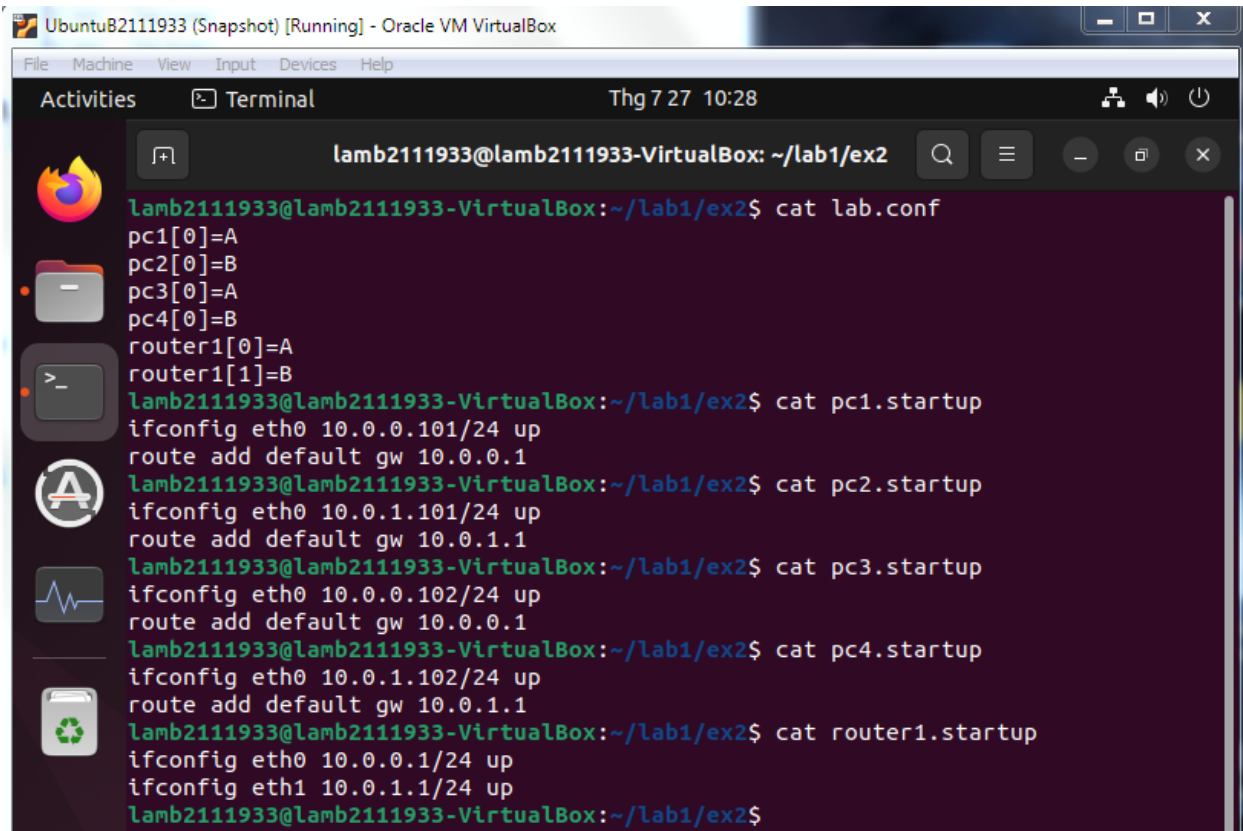
```
$ gedit pc4.startup
```

```
$ gedit router1.startup
```

```

UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thg 7 27 10:18
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ mkdir pc1 pc2 pc3 pc4 router1 shared
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ gedit lab.conf
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ gedit pc1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ gedit pc2.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ gedit pc3.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ gedit pc4.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ gedit router1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ ls
lab.conf  pc1.startup  pc2.startup  pc3.startup  pc4.startup  router1.startup
pc1       pc2         pc3         pc4         router1      shared
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$

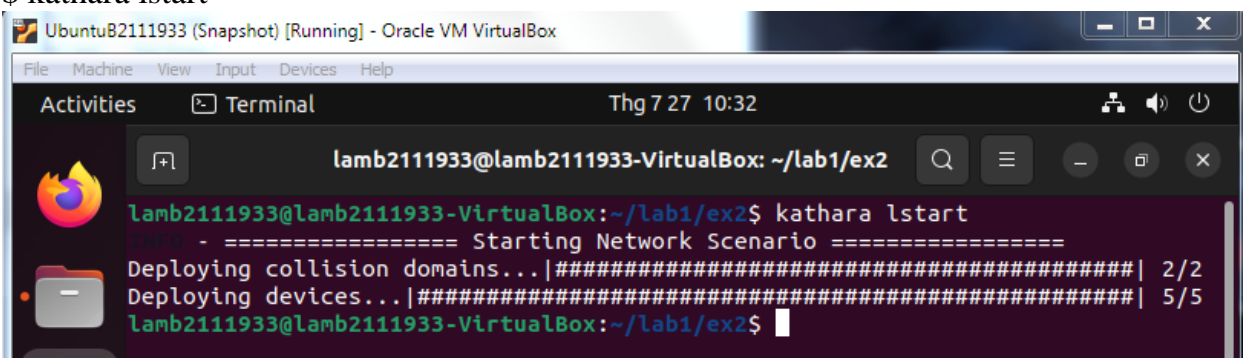
```



```
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ cat lab.conf
pc1[0]=A
pc2[0]=B
pc3[0]=A
pc4[0]=B
router1[0]=A
router1[1]=B
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ cat pc1.startup
ifconfig eth0 10.0.0.101/24 up
route add default gw 10.0.0.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ cat pc2.startup
ifconfig eth0 10.0.1.101/24 up
route add default gw 10.0.1.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ cat pc3.startup
ifconfig eth0 10.0.0.102/24 up
route add default gw 10.0.0.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ cat pc4.startup
ifconfig eth0 10.0.1.102/24 up
route add default gw 10.0.1.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ cat router1.startup
ifconfig eth0 10.0.0.1/24 up
ifconfig eth1 10.0.1.1/24 up
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$
```

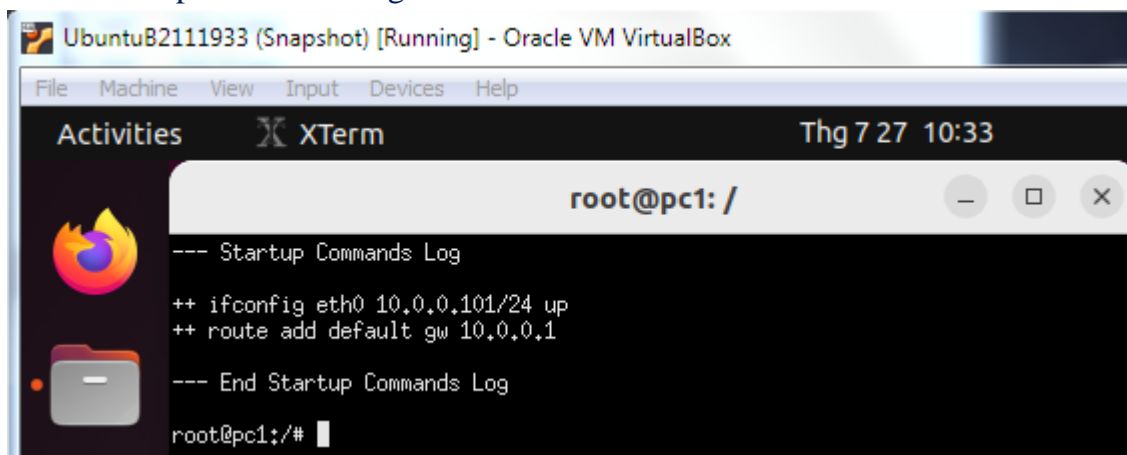
Start the prepared lab:

\$ kathara lstart



```
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ kathara lstart
INFO - ===== Starting Network Scenario =====
Deploying collision domains...|#####| 2/2
Deploying devices...|#####| 5/5
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$
```

It will startup commands log:



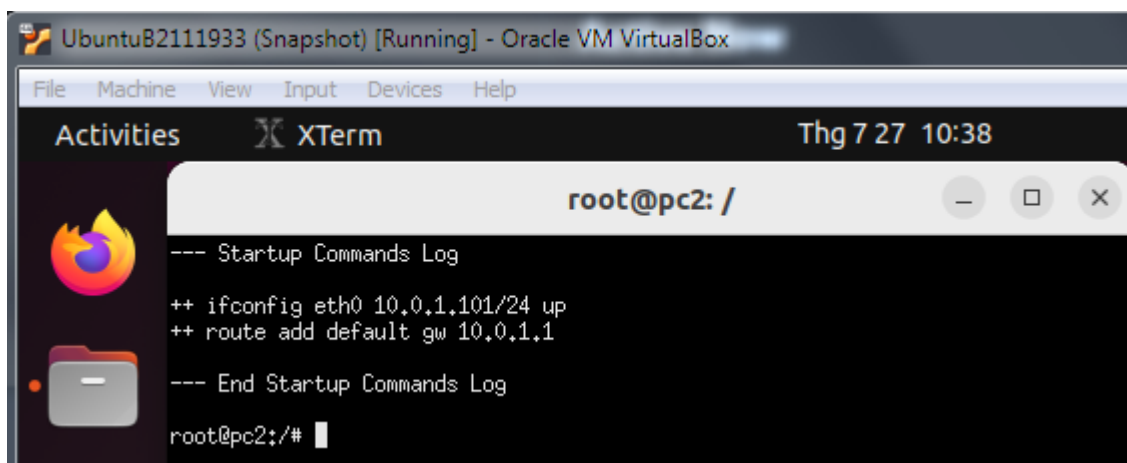
UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities XTerm Thg 7 27 10:33

root@pc1: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.101/24 up
++ route add default gw 10.0.0.1
--- End Startup Commands Log
root@pc1:/#
```



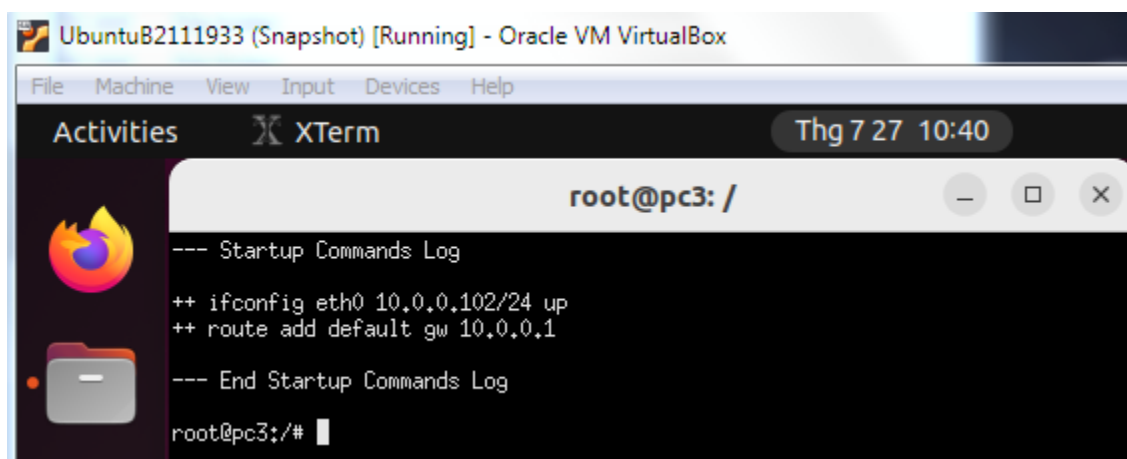
UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities XTerm Thg 7 27 10:38

root@pc2: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.1.101/24 up
++ route add default gw 10.0.1.1
--- End Startup Commands Log
root@pc2:/#
```



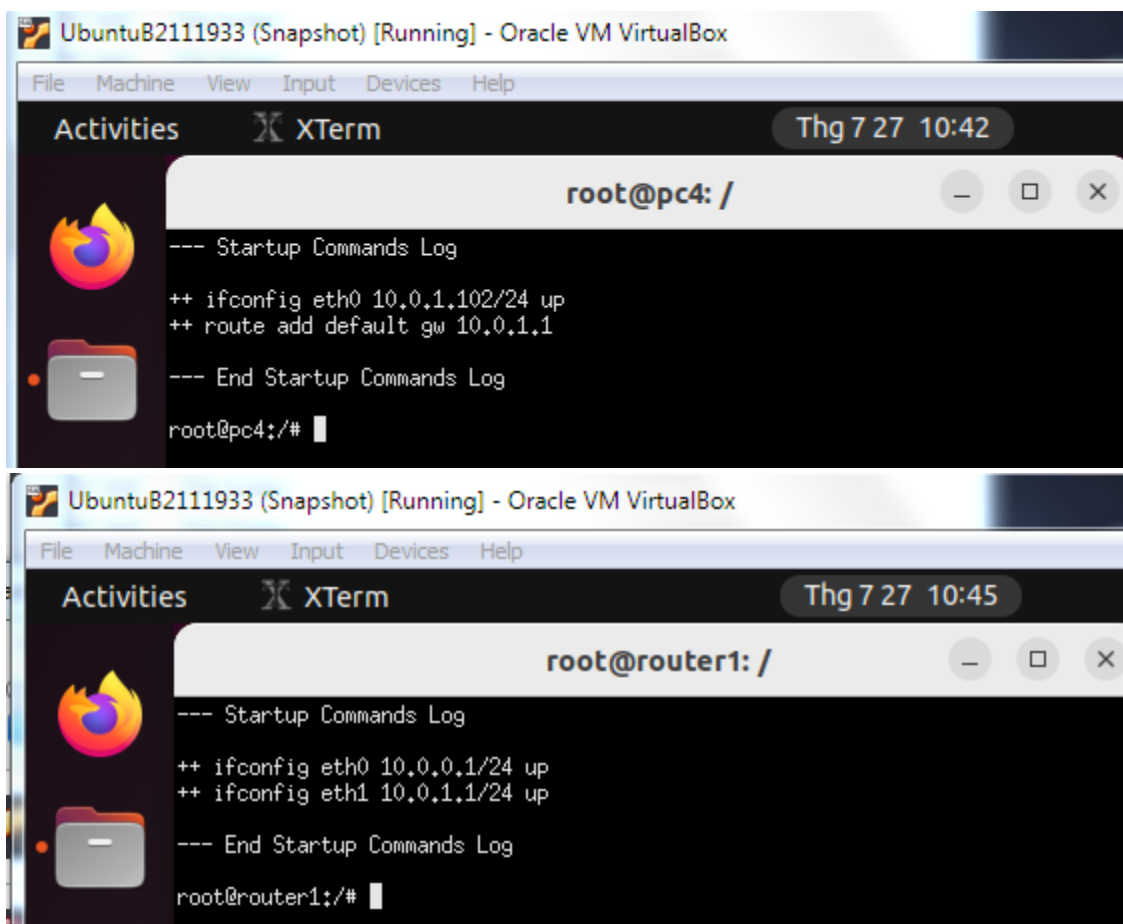
UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

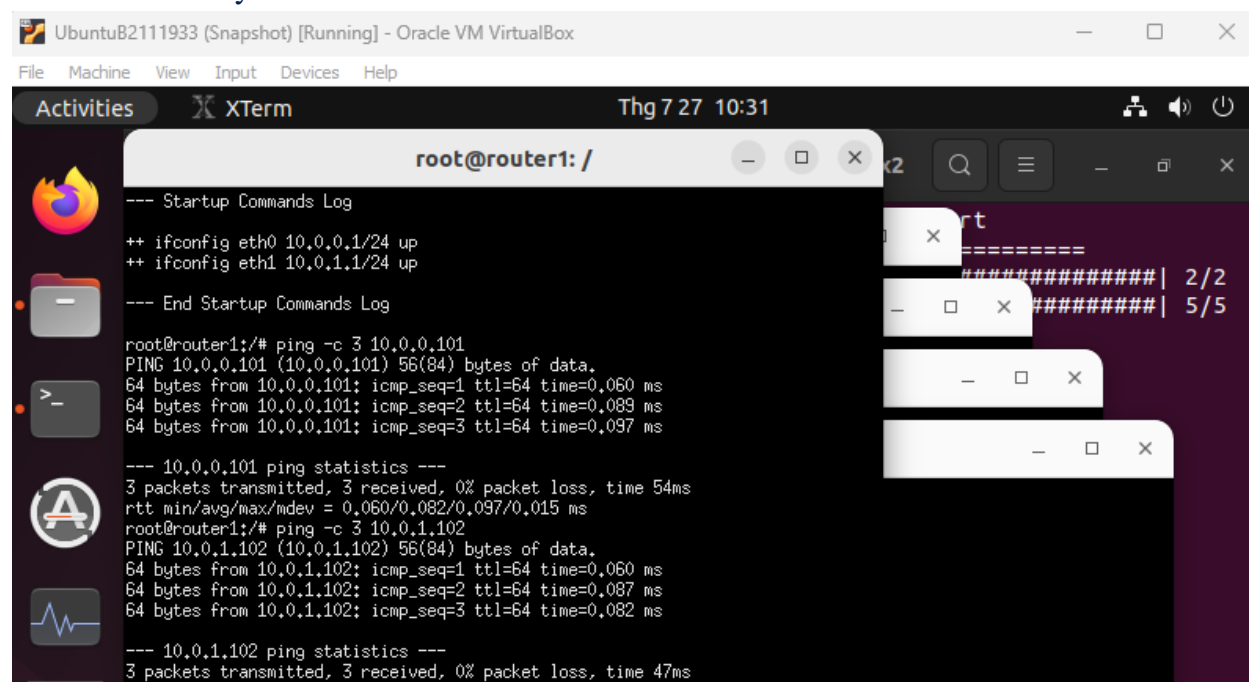
Activities XTerm Thg 7 27 10:40

root@pc3: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.102/24 up
++ route add default gw 10.0.0.1
--- End Startup Commands Log
root@pc3:/#
```



Test connectivity:




```

root@pc3: /
--- Startup Commands Log
++ ifconfig eth0 10.0.0.102/24 up
++ route add default gw 10.0.0.1
--- End Startup Commands Log

root@pc3:/# ping -c 3 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=0.049 ms

--- 10.0.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 58ms
rtt min/avg/max/mdev = 0.033/0.048/0.062/0.011 ms
root@pc3:/# ping -c 3 10.0.1.102
PING 10.0.1.102 (10.0.1.102) 56(84) bytes of data.
64 bytes from 10.0.1.102: icmp_seq=1 ttl=63 time=0.153 ms

```

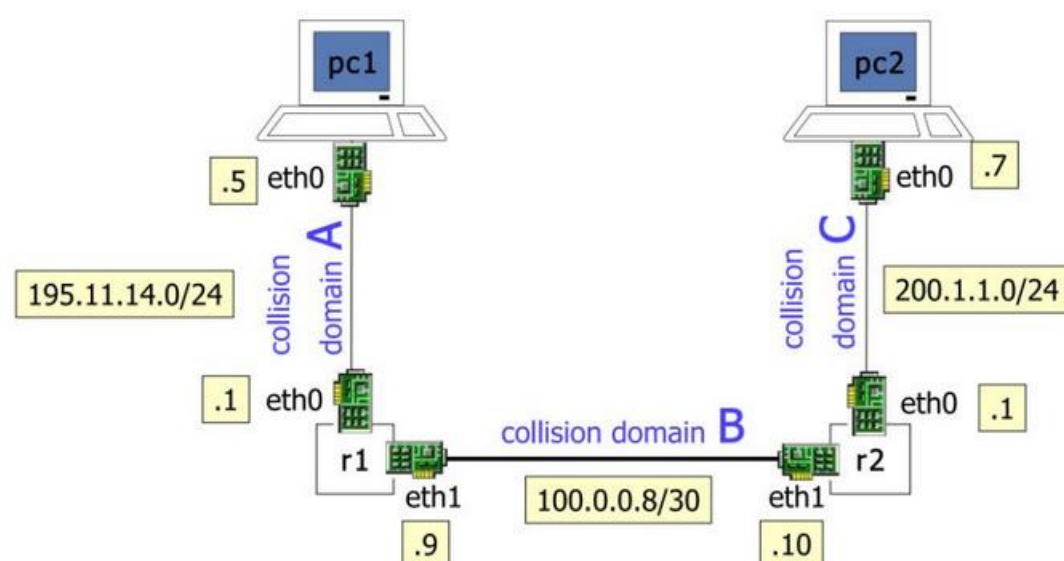
Before leaving, remember to halt all the devices of this lab:

```

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex2$ kathara lclean
===== Stopping Network Scenario =====
Deleting devices...|#####| 5/5
Deleting collision domains...|#####| 2/2

```

Exercise 3: Construct the network



Answer:Prepare a lab for **Exercise 3**:

```
$ mkdir pc1 pc2 router1 router2 shared
```

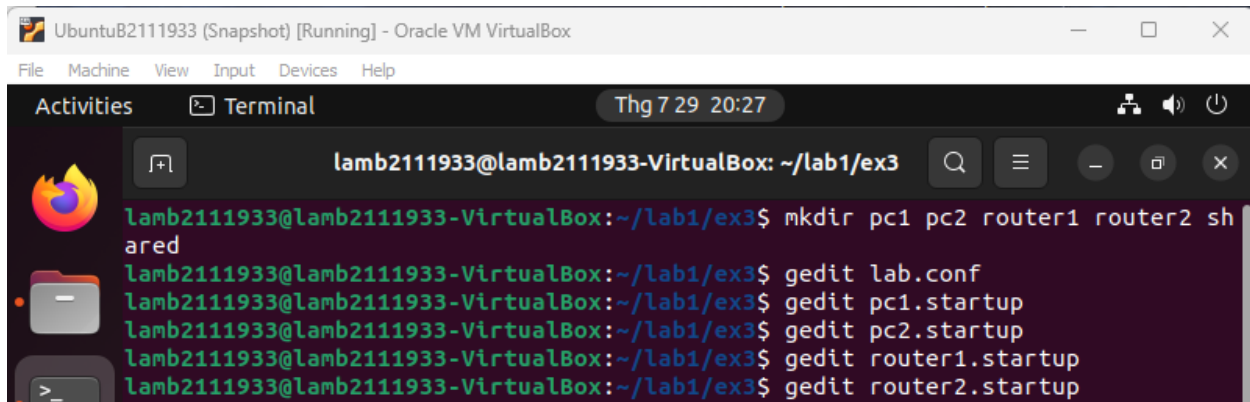
```
$ gedit lab.conf
```

```
$ gedit pc1.startup
```

```
$ gedit pc2.startup
```

```
$ gedit router1.startup
```

```
$ gedit router2.startup
```



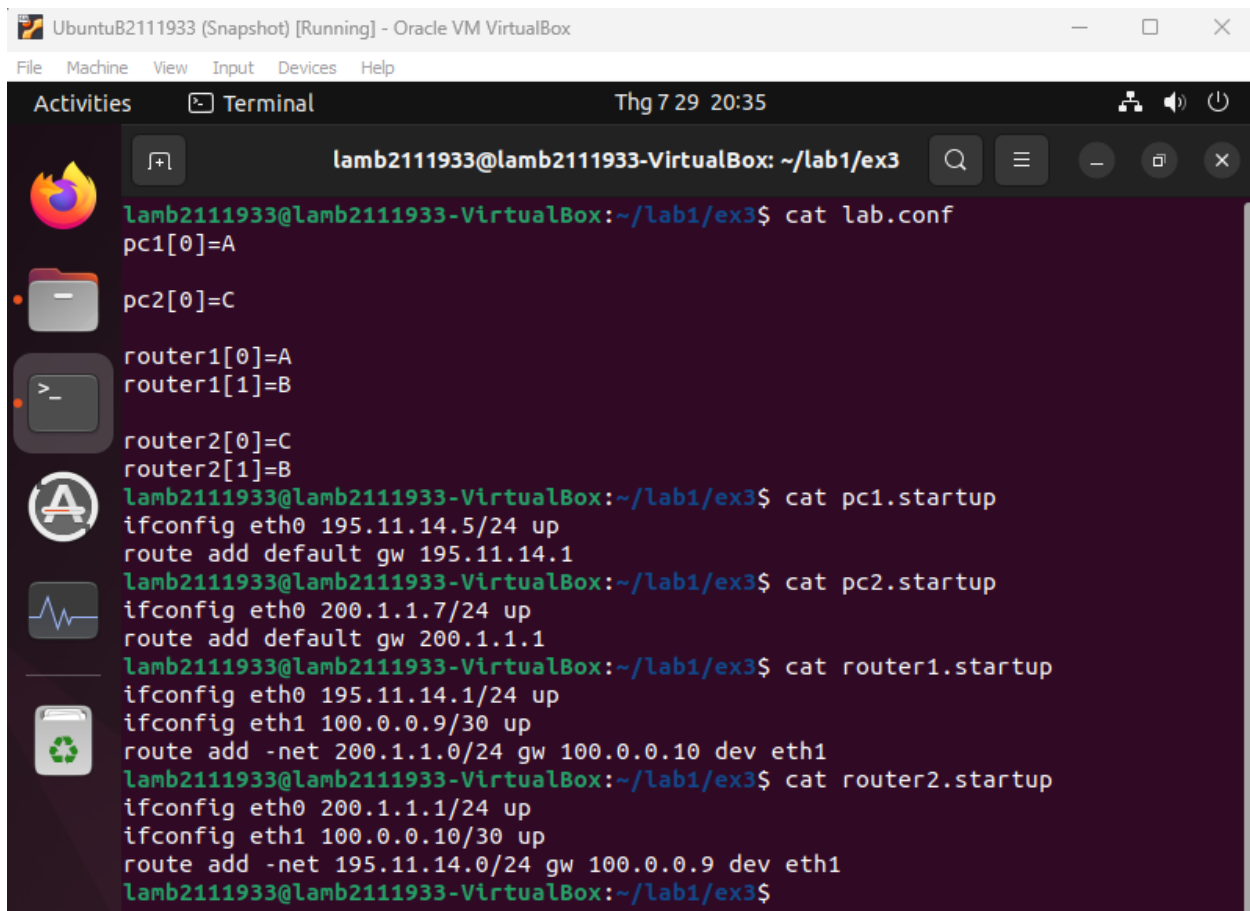
UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Thg 7 29 20:27

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex3

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ mkdir pc1 pc2 router1 router2 shared
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ gedit lab.conf
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ gedit pc1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ gedit pc2.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ gedit router1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ gedit router2.startup
```



UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox

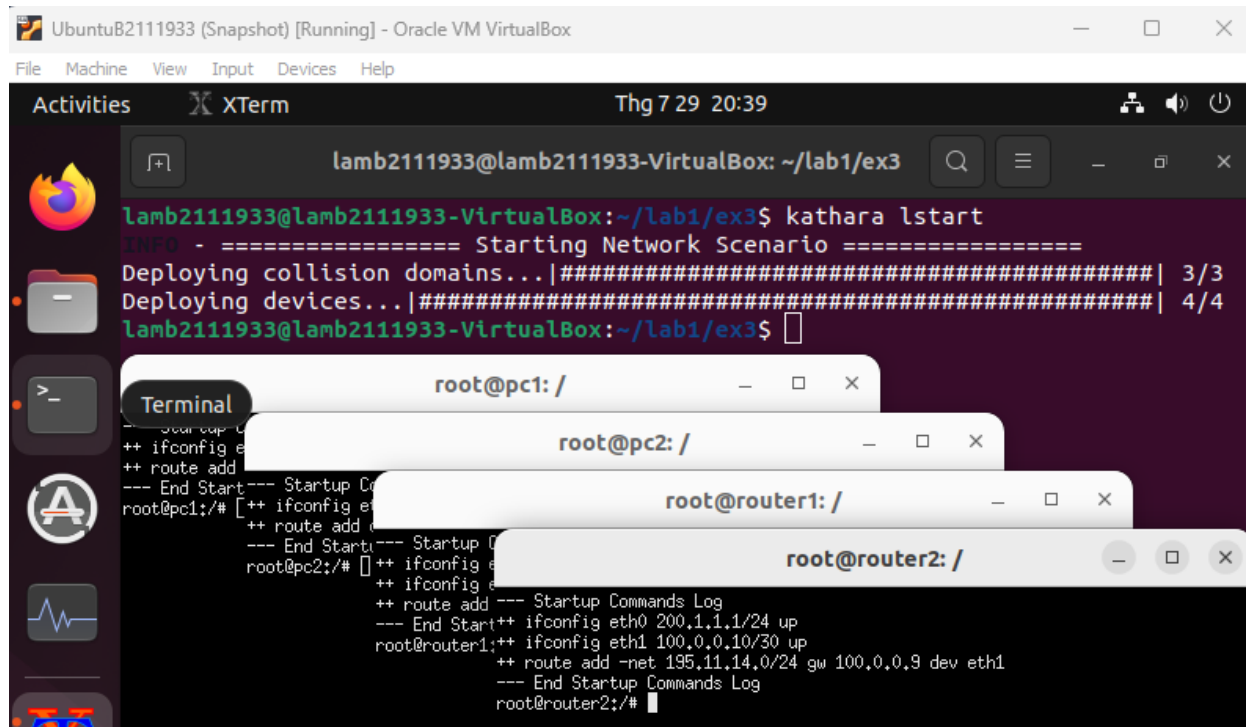
File Machine View Input Devices Help

Activities Terminal Thg 7 29 20:35

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex3

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ cat lab.conf
pc1[0]=A
pc2[0]=C
router1[0]=A
router1[1]=B
router2[0]=C
router2[1]=B
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ cat pc1.startup
ifconfig eth0 195.11.14.5/24 up
route add default gw 195.11.14.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ cat pc2.startup
ifconfig eth0 200.1.1.7/24 up
route add default gw 200.1.1.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ cat router1.startup
ifconfig eth0 195.11.14.1/24 up
ifconfig eth1 100.0.0.9/30 up
route add -net 200.1.1.0/24 gw 100.0.0.10 dev eth1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ cat router2.startup
ifconfig eth0 200.1.1.1/24 up
ifconfig eth1 100.0.0.10/30 up
route add -net 195.11.14.0/24 gw 100.0.0.9 dev eth1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$
```

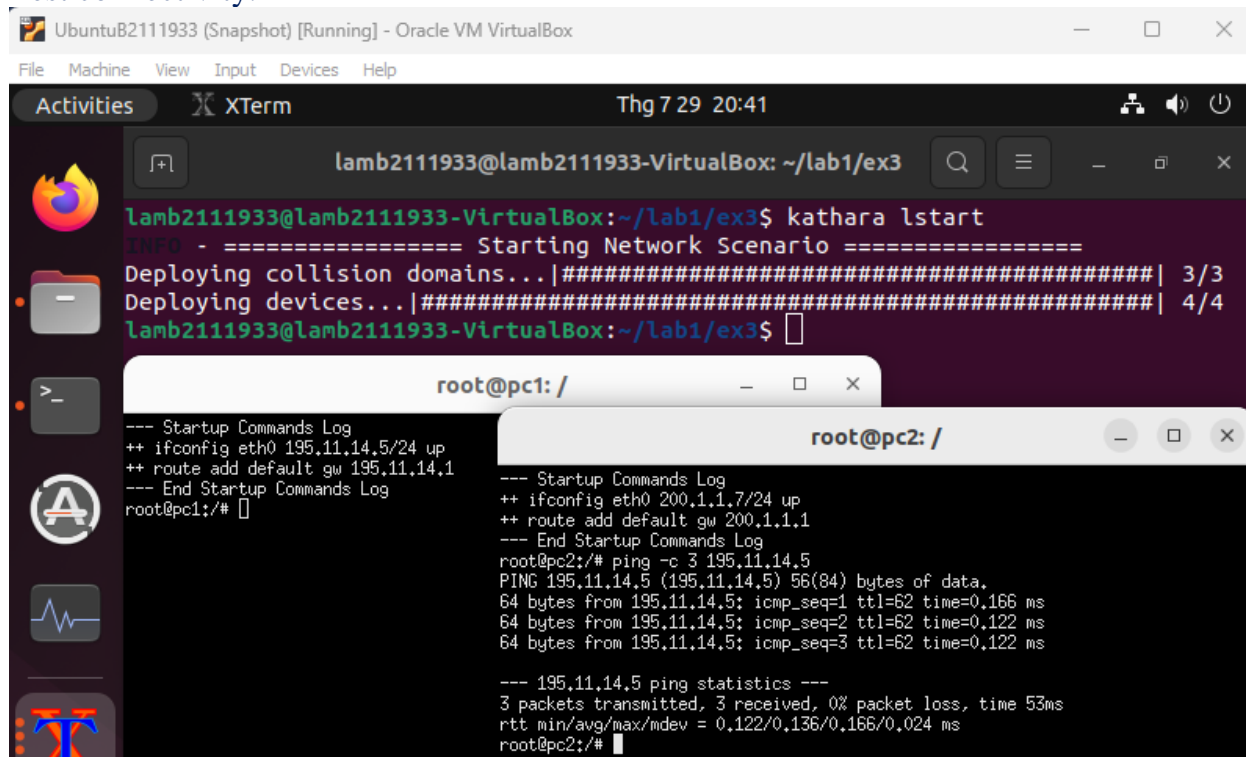
Start the prepared lab:



```
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex3
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ kathara lstart
INFO - ===== Starting Network Scenario =====
Deploying collision domains...|#####| 3/3
Deploying devices...|#####| 4/4
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$

root@pc1: /
root@pc2: /
root@router1: /
root@router2: /
```

Test connectivity:



```
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex3
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ kathara lstart
INFO - ===== Starting Network Scenario =====
Deploying collision domains...|#####| 3/3
Deploying devices...|#####| 4/4
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$

root@pc1: /
--- Startup Commands Log
++ ifconfig eth0 195.11.14.5/24 up
++ route add default gw 195.11.14.1
--- End Startup Commands Log
root@pc1:/#

root@pc2: /
--- Startup Commands Log
++ ifconfig eth0 200.1.1.7/24 up
++ route add default gw 200.1.1.1
--- End Startup Commands Log
root@pc2:/# ping -c 3 195.11.14.5
PING 195.11.14.5 (195.11.14.5) 56(84) bytes of data:
64 bytes from 195.11.14.5: icmp_seq=1 ttl=62 time=0.166 ms
64 bytes from 195.11.14.5: icmp_seq=2 ttl=62 time=0.122 ms
64 bytes from 195.11.14.5: icmp_seq=3 ttl=62 time=0.122 ms

--- 195.11.14.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 53ms
rtt min/avg/max/ndev = 0.122/0.136/0.166/0.024 ms
root@pc2:/#
```

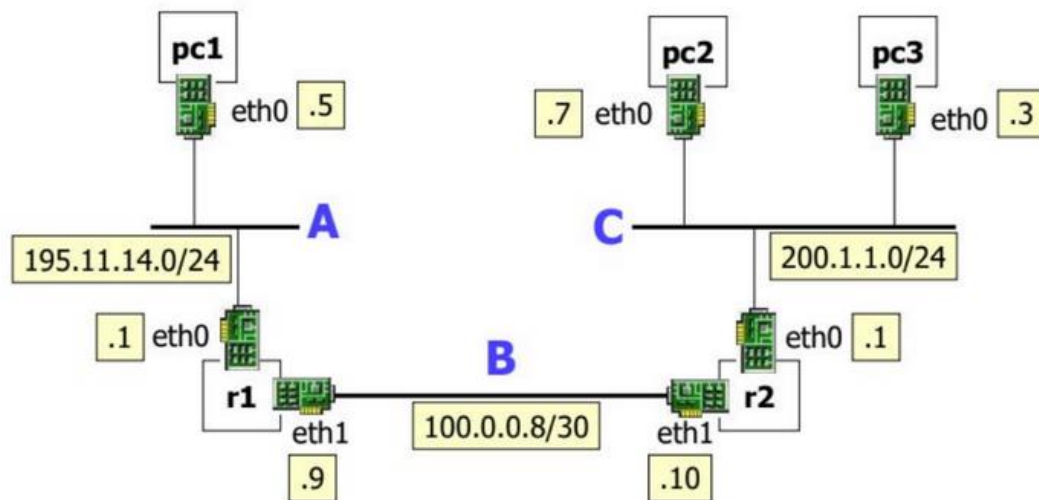
Before leaving, remember to halts all the devices of this lab:

```

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex3
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$ kathara lclean
kathara - ===== Stopping Network Scenario =====
Deleting devices...|#####| 4/4
Deleting collision domains...|#####| 3/3
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex3$

```

Exercise 4: study arp protocol



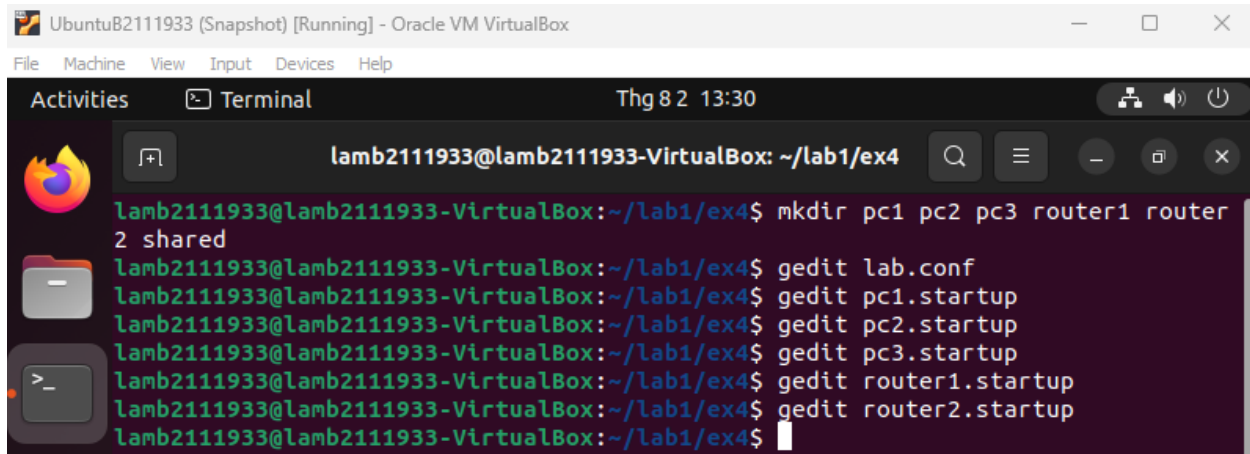
Answer:

Prepare a lab for **Exercise 4**:

```

$ mkdir pc1 pc2 pc3 router1 router2 shared
$ gedit lab.conf
$ gedit pc1.startup
$ gedit pc2.startup
$ gedit pc3.startup
$ gedit router1.startup
$ gedit router2.startup

```



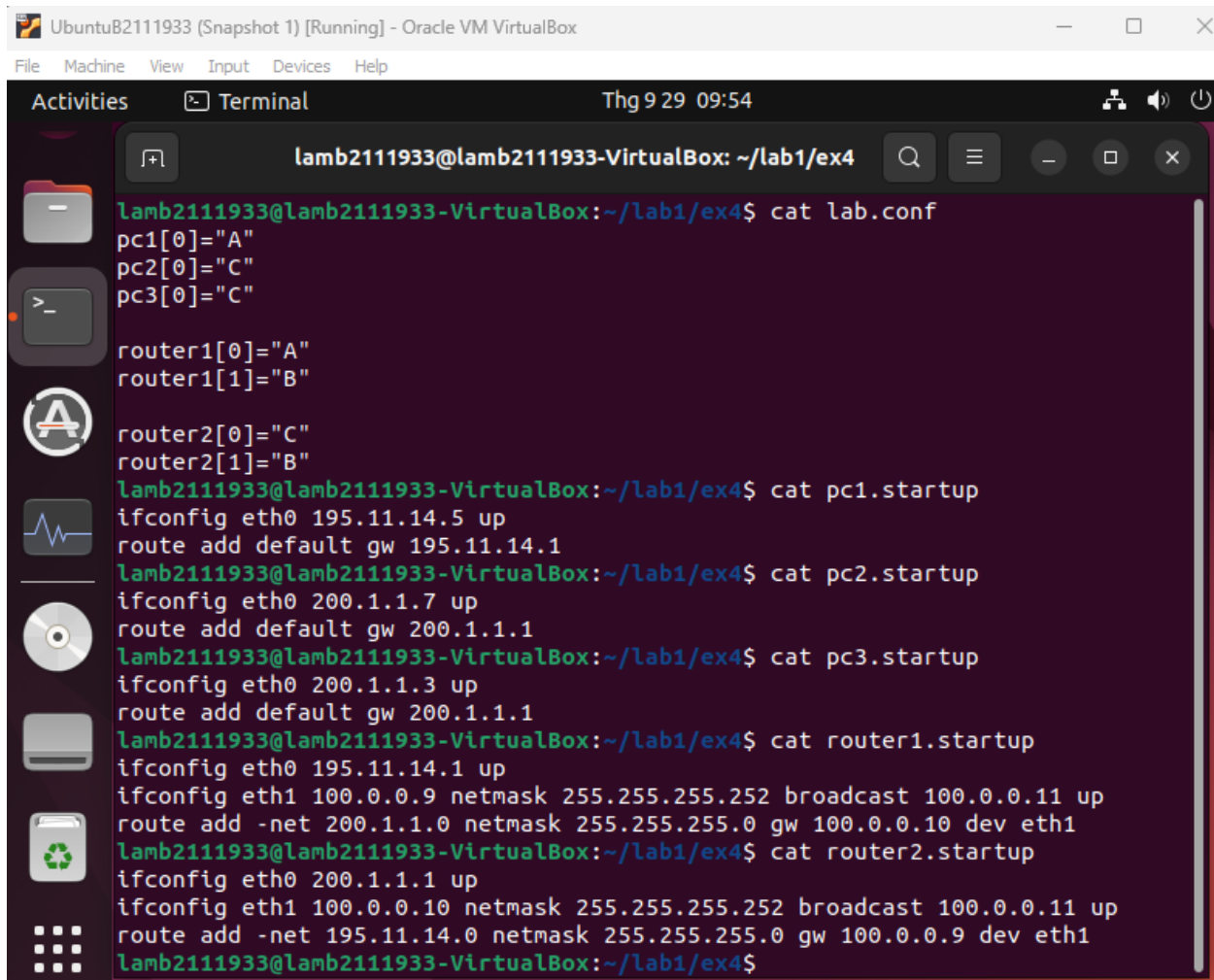
UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Thg 8 2 13:30

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex4

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ mkdir pc1 pc2 pc3 router1 router2 shared
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ gedit lab.conf
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ gedit pc1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ gedit pc2.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ gedit pc3.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ gedit router1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ gedit router2.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$
```



UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

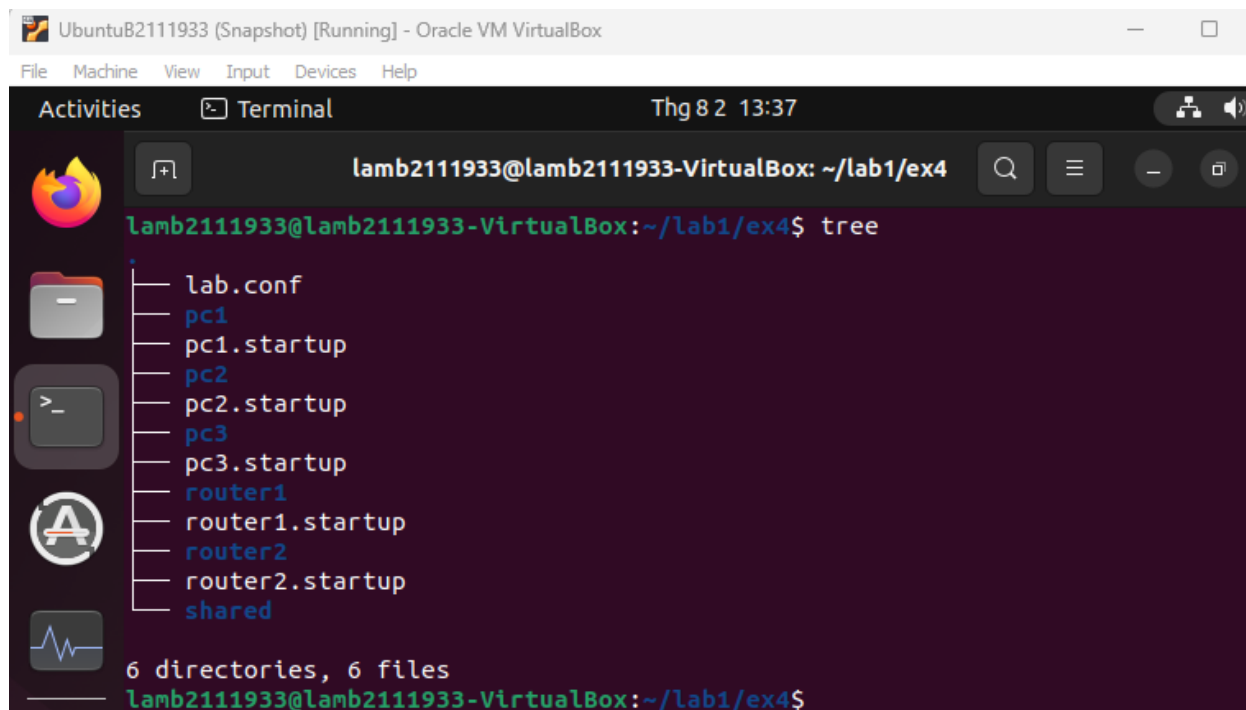
Activities Terminal Thg 9 29 09:54

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex4

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ cat lab.conf
pc1[0]="A"
pc2[0]="C"
pc3[0]="C"

router1[0]="A"
router1[1]="B"

router2[0]="C"
router2[1]="B"
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ cat pc1.startup
ifconfig eth0 195.11.14.5 up
route add default gw 195.11.14.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ cat pc2.startup
ifconfig eth0 200.1.1.7 up
route add default gw 200.1.1.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ cat pc3.startup
ifconfig eth0 200.1.1.3 up
route add default gw 200.1.1.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ cat router1.startup
ifconfig eth0 195.11.14.1 up
ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ cat router2.startup
ifconfig eth0 200.1.1.1 up
ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$
```



UbuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

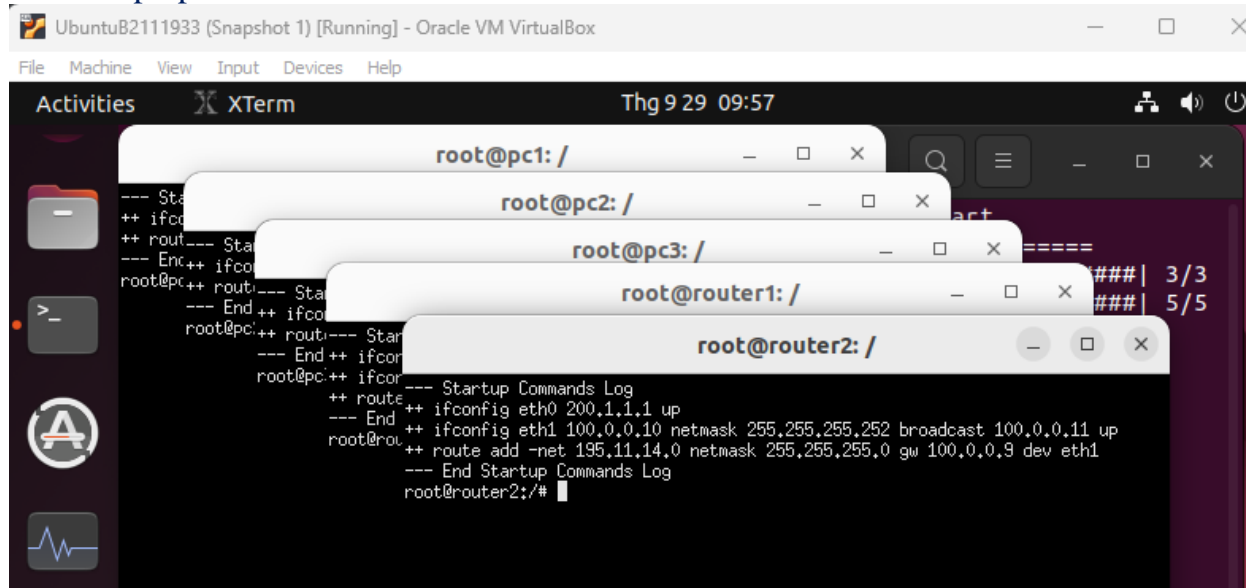
Activities Terminal Thg 8 2 13:37

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex4

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ tree
.
├── lab.conf
├── pc1
├── pc1.startup
├── pc2
├── pc2.startup
├── pc3
├── pc3.startup
├── router1
├── router1.startup
├── router2
├── router2.startup
└── shared

6 directories, 6 files
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$
```

Start the prepared lab:



UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities XTerm Thg 9 29 09:57

root@pc1: /

root@pc2: /

root@pc3: /

root@router1: /

root@router2: /

```
--- Start
++ ifconfig eth0 200.1.1.1 up
--- End
root@pc1: /
```

```
--- Start
++ ifconfig eth0 200.1.1.1 up
--- End
root@pc2: /
```

```
--- Start
++ ifconfig eth0 200.1.1.1 up
--- End
root@pc3: /
```

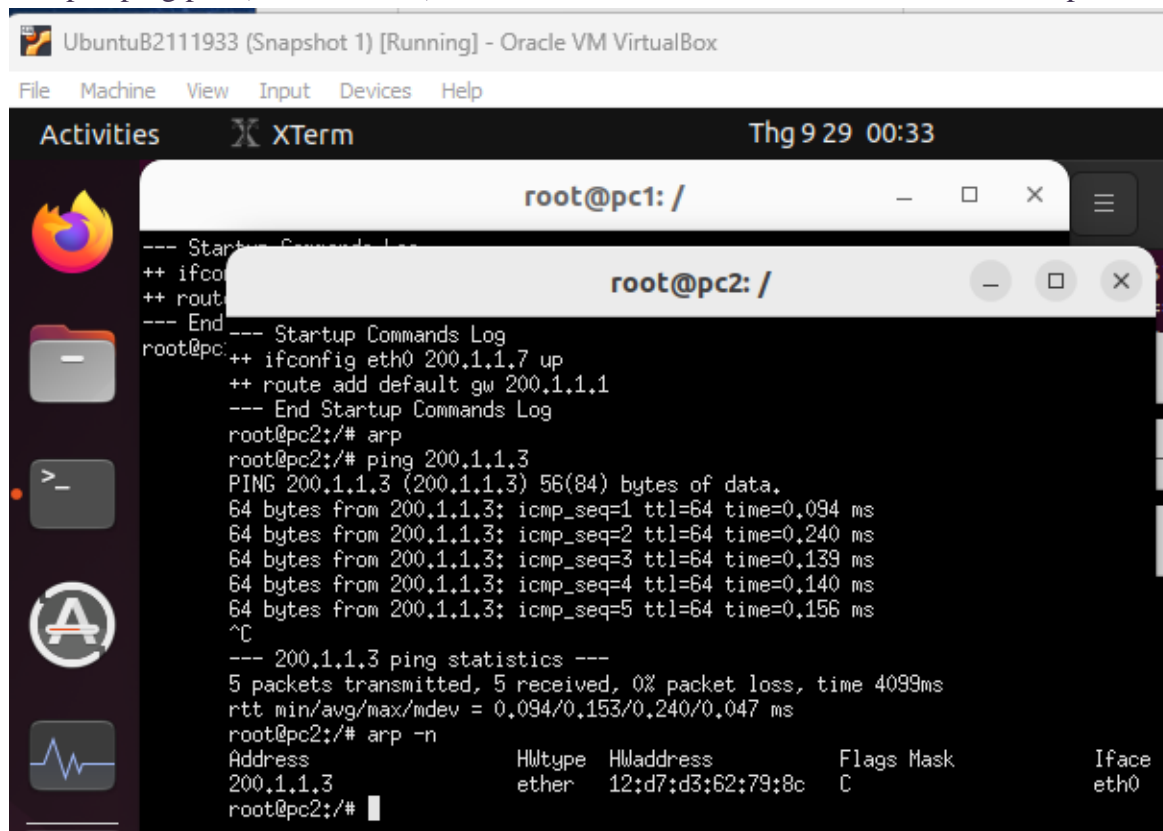
```
--- Start
++ ifconfig eth0 200.1.1.1 up
--- End
root@router1: /
```

```
--- Start
++ ifconfig eth0 200.1.1.1 up
--- End
root@router2: /
```

```
--- Startup Commands Log
++ route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
--- End Startup Commands Log
root@router2: /#
```

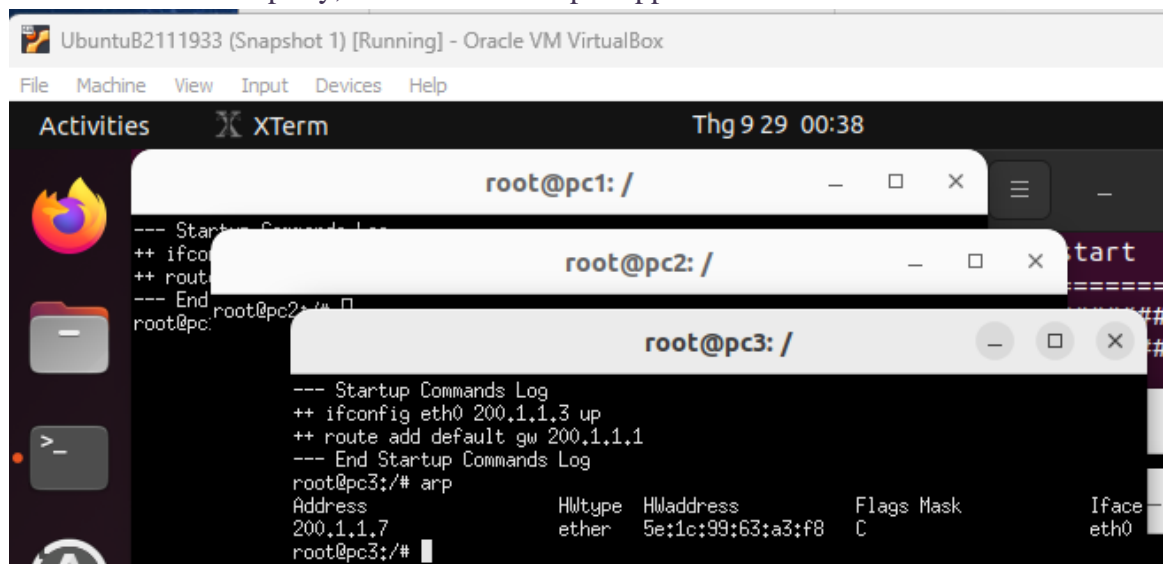
Inspect the arp cache (local traffic):

Use pc2 ping pc3 (local network) => Address resolution results are stored in the arp cache



```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.7 up
++ route add default gw 200.1.1.1
--- End Startup Commands Log
root@pc1:~#
root@pc2:~# arp
root@pc2:~# ping 200.1.1.3
PING 200.1.1.3 (200.1.1.3) 56(84) bytes of data:
64 bytes from 200.1.1.3: icmp_seq=1 ttl=64 time=0.094 ms
64 bytes from 200.1.1.3: icmp_seq=2 ttl=64 time=0.240 ms
64 bytes from 200.1.1.3: icmp_seq=3 ttl=64 time=0.139 ms
64 bytes from 200.1.1.3: icmp_seq=4 ttl=64 time=0.140 ms
64 bytes from 200.1.1.3: icmp_seq=5 ttl=64 time=0.156 ms
^C
--- 200.1.1.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.094/0.153/0.240/0.047 ms
root@pc2:~# arp -n
Address                  Hwtype  Hwaddress      Flags Mask      Iface
200.1.1.3                ether   12:d7:d3:62:79:8c  C                eth0
root@pc2:~#
```

Communications are usually bi-directional. The receiver of the arp request learns the mac address of the other party, to avoid a new arpin opposite direction.

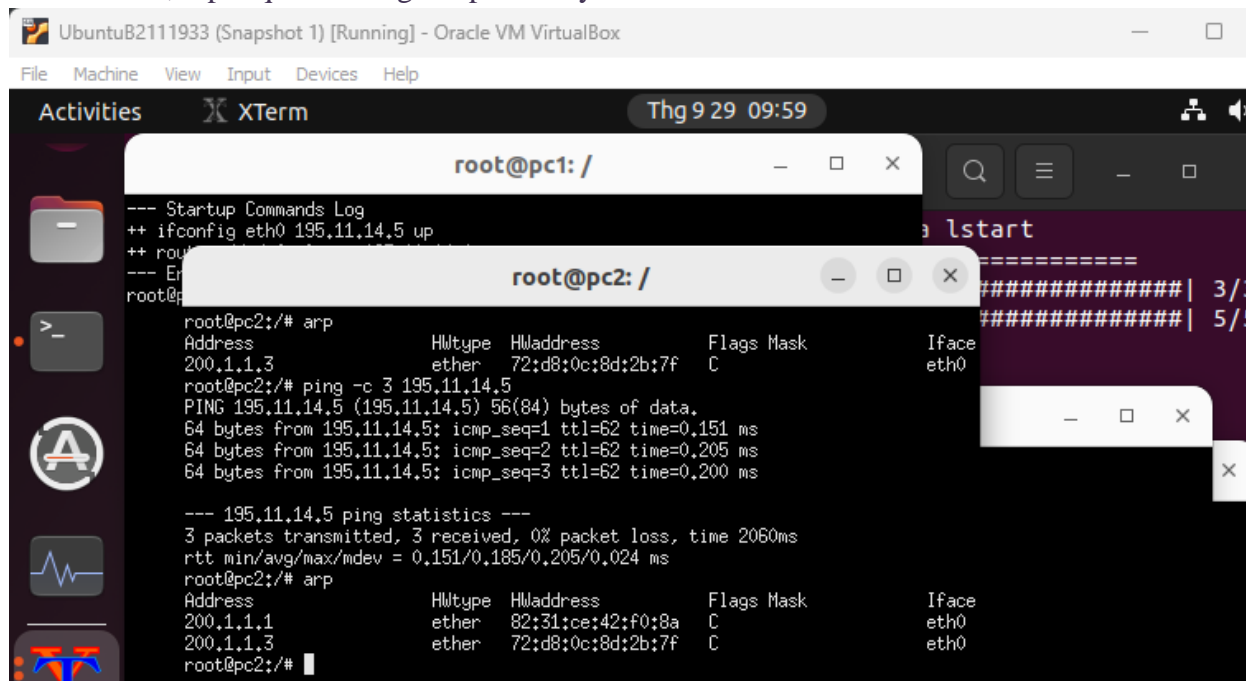


```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.3 up
++ route add default gw 200.1.1.1
--- End Startup Commands Log
root@pc3:~# arp
root@pc3:~#
root@pc3:~# arp -n
Address                  Hwtype  Hwaddress      Flags Mask      Iface
200.1.1.7                ether   5e:1c:99:63:a3:f8  C                eth0
root@pc3:~#
```


Inspect the arp cache (non local traffic):

Use pc2 ping pc1 (non local network)

=> When ip traffic is addressed outside the local network, the sender needs the mac address of the router, arp requests can get replies only within the local network.



```

root@pc1: /
--- Startup Commands Log
++ ifconfig eth0 195.11.14.5 up
++ route add -net 195.11.14.0 netmask 255.255.255.0 gw 195.11.14.1 dev eth0
--- End Startup Commands Log

root@pc2: /
root@pc2: /# arp
Address          HWtype  HWaddress      Flags Mask    Iface
200.1.1.3         ether    72:d8:0c:8d:2b:7f  C           eth0

root@pc2: /# ping -c 3 195.11.14.5
PING 195.11.14.5 (195.11.14.5) 56(84) bytes of data:
64 bytes from 195.11.14.5: icmp_seq=1 ttl=62 time=0.151 ms
64 bytes from 195.11.14.5: icmp_seq=2 ttl=62 time=0.205 ms
64 bytes from 195.11.14.5: icmp_seq=3 ttl=62 time=0.200 ms

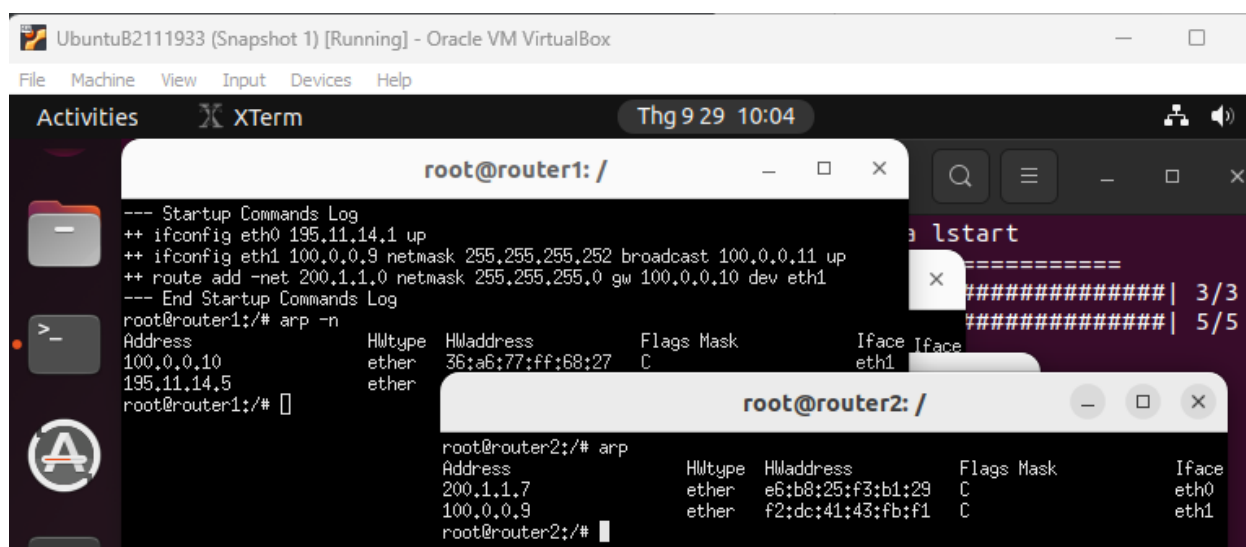
--- 195.11.14.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2060ms
rtt min/avg/max/mdev = 0.151/0.185/0.205/0.024 ms

root@pc2: /# arp
Address          HWtype  HWaddress      Flags Mask    Iface
200.1.1.1         ether    82:31:ce:42:f0:8a  C           eth0
200.1.1.3         ether    72:d8:0c:8d:2b:7f  C           eth0
root@pc2: /#

```

Inspect arp of the router:

- Routers perform arp too (hence have arp caches) anytime they have to send ip packets on an ethernet LAN.



```

root@router1: /
--- Startup Commands Log
++ ifconfig eth0 195.11.14.1 up
++ ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
++ route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
--- End Startup Commands Log

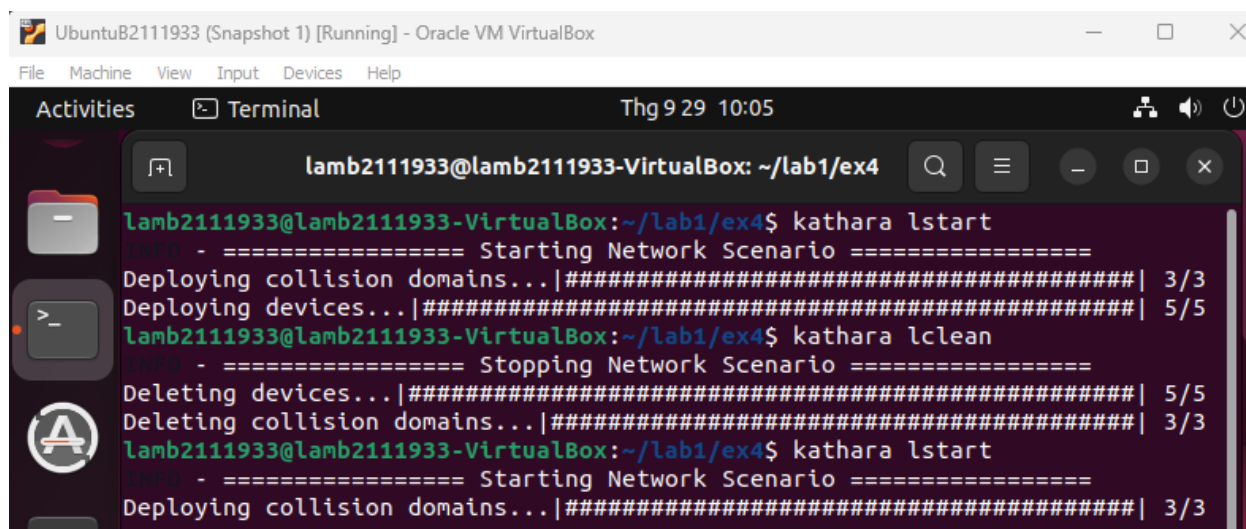
root@router1: /# arp -n
Address          HWtype  HWaddress      Flags Mask    Iface
100.0.0.10        ether    36:a6:77:ff:68:27  C           eth1
195.11.14.5        ether    72:d8:0c:8d:2b:7f  C           eth0
root@router1: /#

root@router2: /
root@router2: /# arp
Address          HWtype  HWaddress      Flags Mask    Iface
200.1.1.7         ether    e6:b8:25:f3:b1:29  C           eth0
100.0.0.9          ether    f2:dc:41:43:fb:f1  C           eth1
root@router2: /#

```

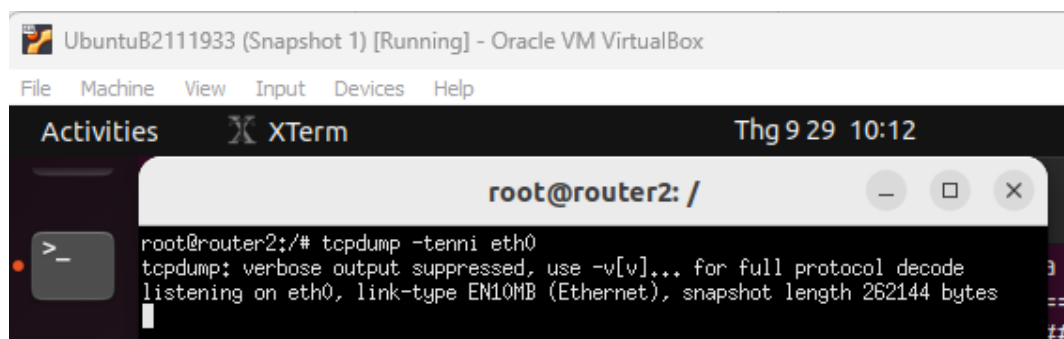

Sniff arp traffic:

Restart the lab in order to clear arp caches

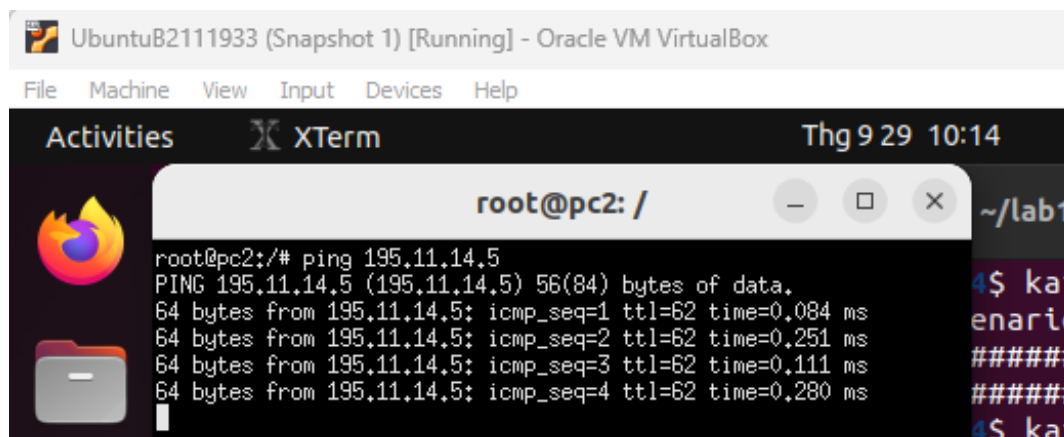


```
lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex4
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ kathara lstart
INFO - ===== Starting Network Scenario =====
Deploying collision domains...|#####| 3/3
Deploying devices...|#####| 5/5
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ kathara lclean
INFO - ===== Stopping Network Scenario =====
Deleting devices...|#####| 5/5
Deleting collision domains...|#####| 3/3
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex4$ kathara lstart
INFO - ===== Starting Network Scenario =====
Deploying collision domains...|#####| 3/3
```

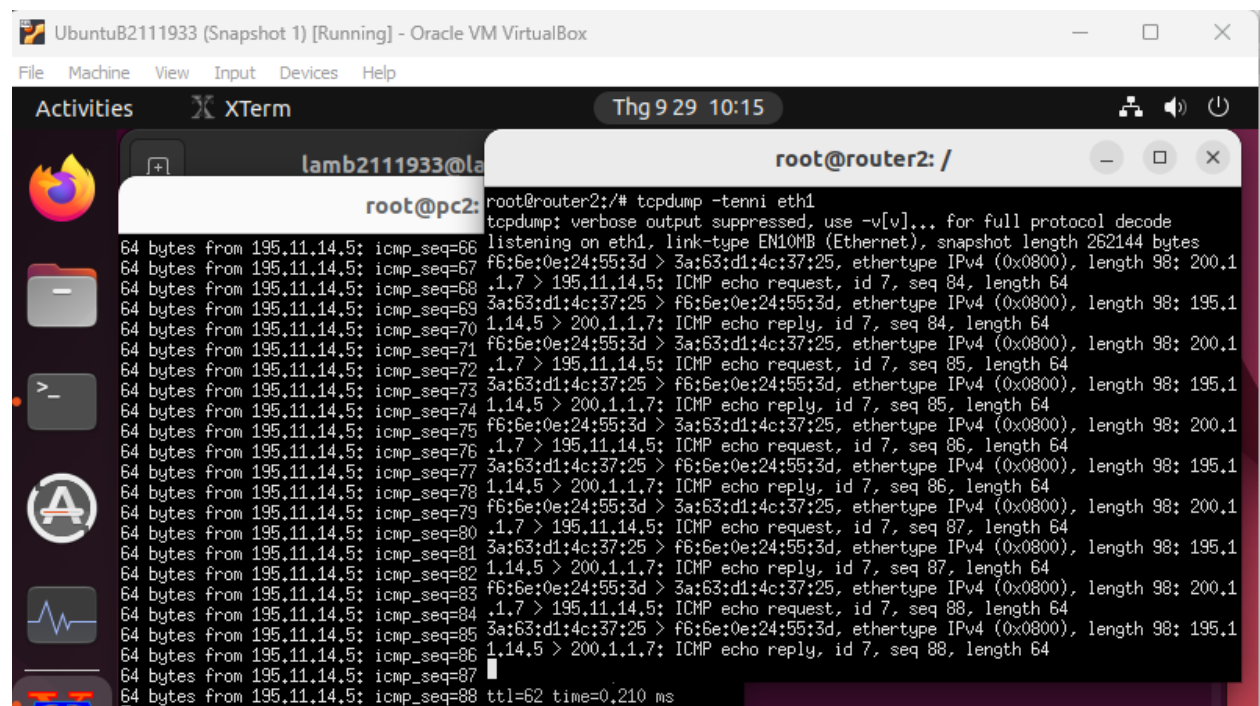
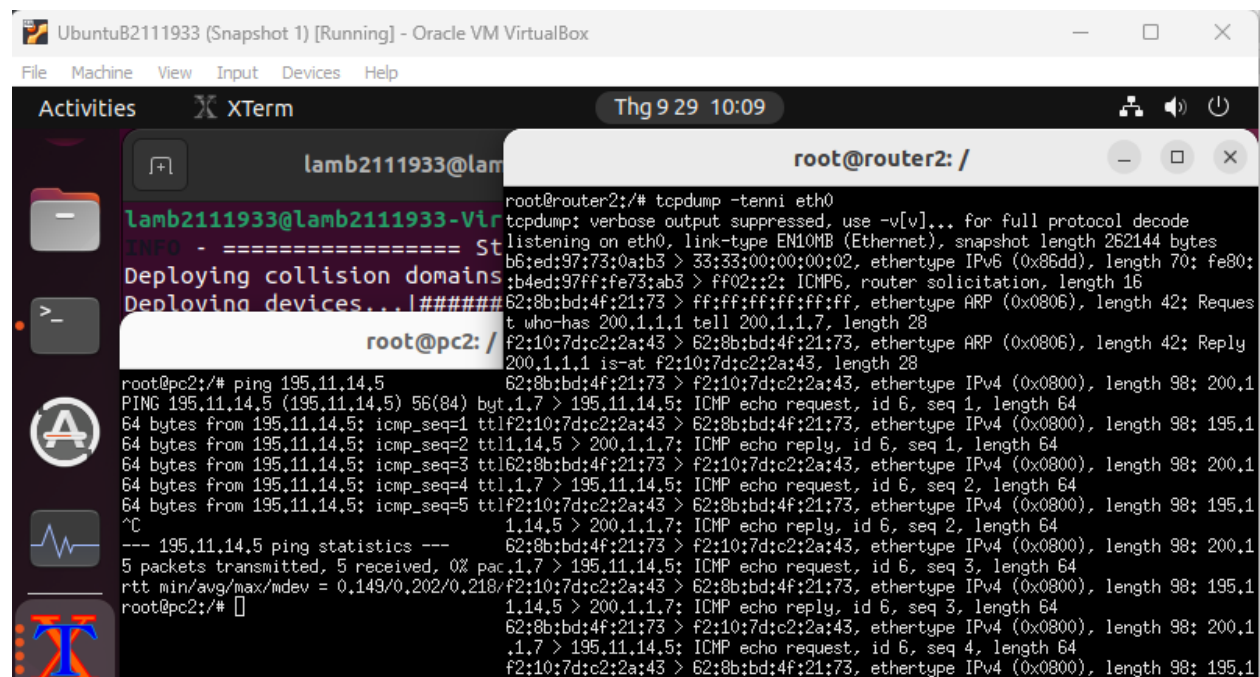
Start to sniff:



```
root@router2: /
root@router2:/# tcpdump -tnni eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```



```
root@pc2: /
root@pc2:/# ping 195.11.14.5
PING 195.11.14.5 (195.11.14.5) 56(84) bytes of data.
64 bytes from 195.11.14.5: icmp_seq=1 ttl=62 time=0.084 ms
64 bytes from 195.11.14.5: icmp_seq=2 ttl=62 time=0.251 ms
64 bytes from 195.11.14.5: icmp_seq=3 ttl=62 time=0.111 ms
64 bytes from 195.11.14.5: icmp_seq=4 ttl=62 time=0.280 ms
```



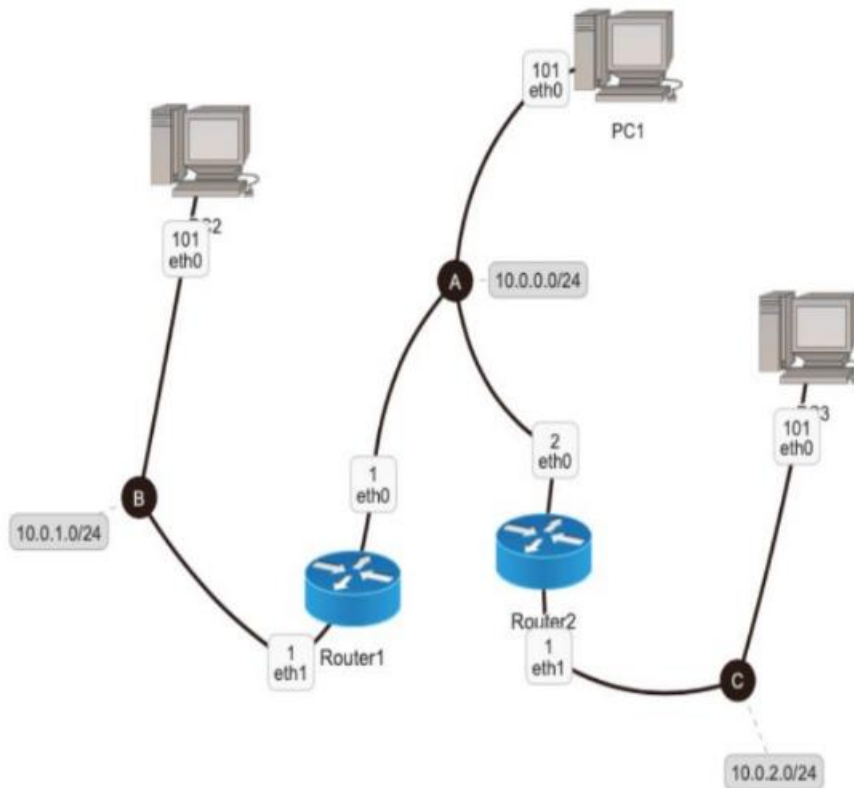
On collision domain A:

```

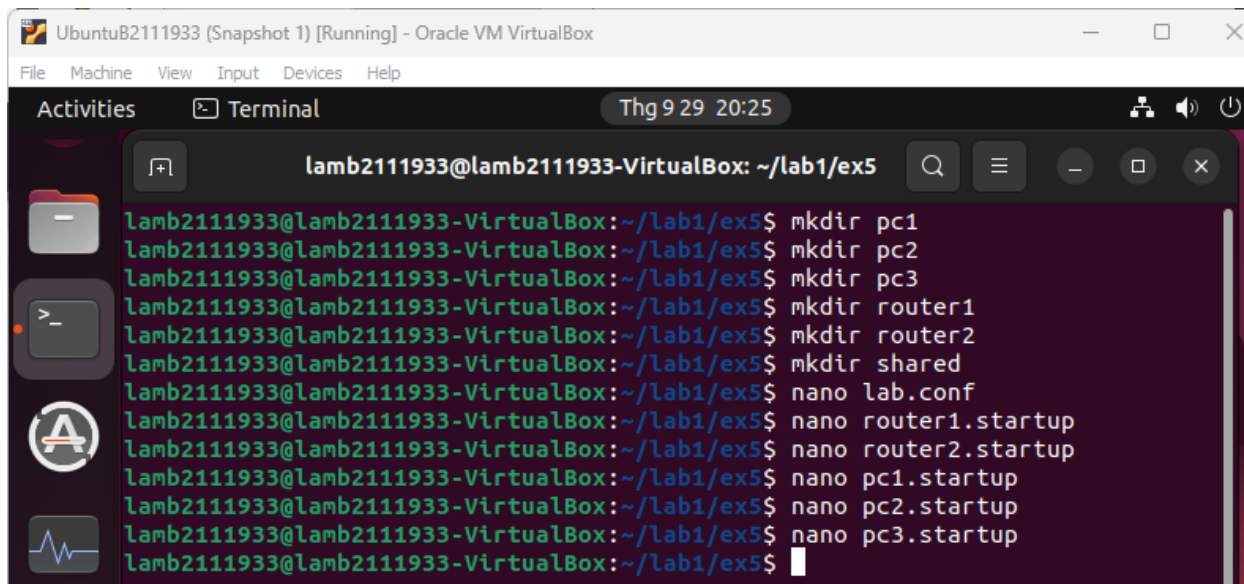
root@pc1: /# tcpdump -ttni eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EM10MB (Ethernet), snapshot length 262144 bytes
64 bytes from 195.11.14.5: icmp_seq=136 ca:d5:bb:bf:ce:73 > 1a:76:41:ec:e6:07, ethertype IPv4 (0x0800), length 98: 200.1
64 bytes from 195.11.14.5: icmp_seq=137 1.7 > 195.11.14.5: ICMP echo request, id 7, seq 154, length 64
64 bytes from 195.11.14.5: icmp_seq=138 1a:76:41:ec:e6:07 > ca:d5:bb:bf:ce:73, ethertype IPv4 (0x0800), length 98: 195.1
64 bytes from 195.11.14.5: icmp_seq=139 1.14.5 > 200.1.1.7: ICMP echo reply, id 7, seq 154, length 64
64 bytes from 195.11.14.5: icmp_seq=140 ca:d5:bb:bf:ce:73 > 1a:76:41:ec:e6:07, ethertype IPv4 (0x0800), length 98: 200.1
64 bytes from 195.11.14.5: icmp_seq=141 1.7 > 195.11.14.5: ICMP echo request, id 7, seq 155, length 64
64 bytes from 195.11.14.5: icmp_seq=142 1a:76:41:ec:e6:07 > ca:d5:bb:bf:ce:73, ethertype IPv4 (0x0800), length 98: 195.1
64 bytes from 195.11.14.5: icmp_seq=143 1.14.5 > 200.1.1.7: ICMP echo reply, id 7, seq 155, length 64
64 bytes from 195.11.14.5: icmp_seq=144 ca:d5:bb:bf:ce:73 > 1a:76:41:ec:e6:07, ethertype IPv4 (0x0800), length 98: 200.1
64 bytes from 195.11.14.5: icmp_seq=145 1.7 > 195.11.14.5: ICMP echo request, id 7, seq 156, length 64
64 bytes from 195.11.14.5: icmp_seq=146 1a:76:41:ec:e6:07 > ca:d5:bb:bf:ce:73, ethertype IPv4 (0x0800), length 98: 195.1
64 bytes from 195.11.14.5: icmp_seq=147 1.14.5 > 200.1.1.7: ICMP echo reply, id 7, seq 156, length 64
64 bytes from 195.11.14.5: icmp_seq=148 ca:d5:bb:bf:ce:73 > 1a:76:41:ec:e6:07, ethertype IPv4 (0x0800), length 98: 200.1
64 bytes from 195.11.14.5: icmp_seq=149 1.7 > 195.11.14.5: ICMP echo request, id 7, seq 157, length 64
64 bytes from 195.11.14.5: icmp_seq=150 1a:76:41:ec:e6:07 > ca:d5:bb:bf:ce:73, ethertype IPv4 (0x0800), length 98: 195.1
64 bytes from 195.11.14.5: icmp_seq=151 1.14.5 > 200.1.1.7: ICMP echo reply, id 7, seq 157, length 64
64 bytes from 195.11.14.5: icmp_seq=152 ca:d5:bb:bf:ce:73 > 1a:76:41:ec:e6:07, ethertype IPv4 (0x0800), length 98: 200.1
64 bytes from 195.11.14.5: icmp_seq=153 1.7 > 195.11.14.5: ICMP echo request, id 7, seq 158, length 64
64 bytes from 195.11.14.5: icmp_seq=154 1a:76:41:ec:e6:07 > ca:d5:bb:bf:ce:73, ethertype IPv4 (0x0800), length 98: 195.1
64 bytes from 195.11.14.5: icmp_seq=155 1.14.5 > 200.1.1.7: ICMP echo reply, id 7, seq 158, length 64
64 bytes from 195.11.14.5: icmp_seq=156 1.14.5 > 200.1.1.7: ICMP echo reply, id 7, seq 158, length 64

```

Exercise 5: Construct the network following the picture



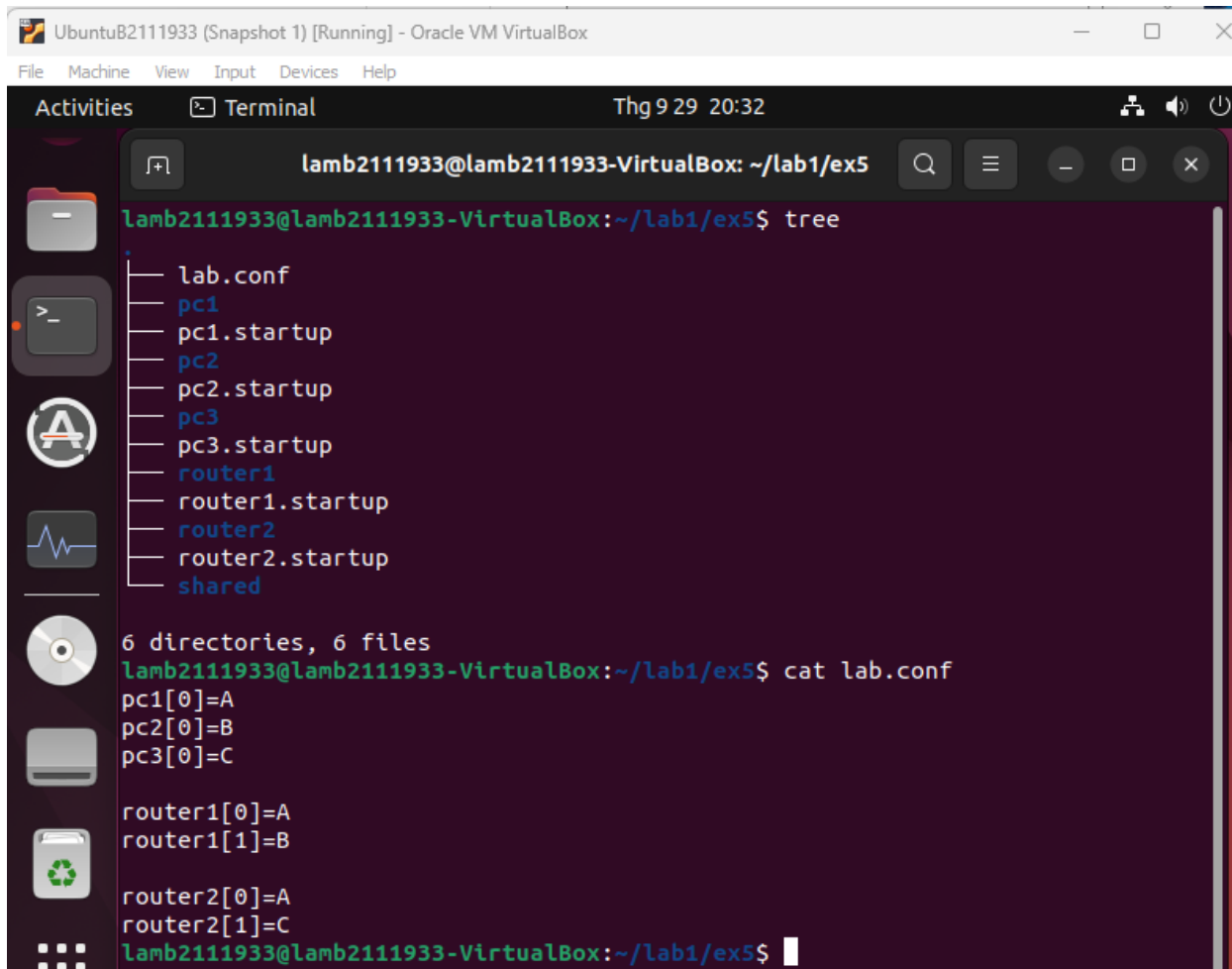
Create folders and files for exercise 5:



The screenshot shows a terminal window titled "UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal prompt is "lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex5". The user has executed the following commands:

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ mkdir pc1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ mkdir pc2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ mkdir pc3
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ mkdir router1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ mkdir router2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ mkdir shared
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ nano lab.conf
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ nano router1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ nano router2.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ nano pc1.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ nano pc2.startup
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ nano pc3.startup
```

Construct the network:



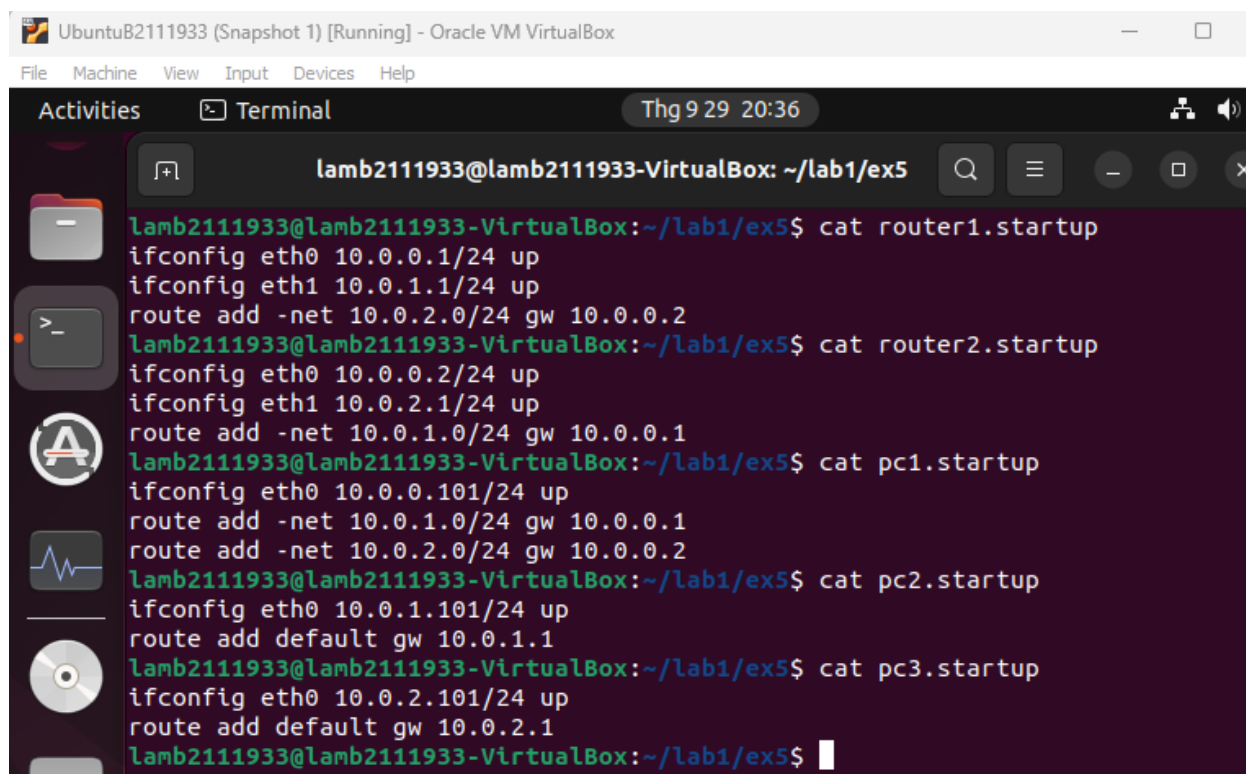
The screenshot shows a terminal window titled "UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal prompt is "lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex5". The user has executed the following commands:

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ tree
.
├── lab.conf
├── pc1
├── pc1.startup
├── pc2
├── pc2.startup
├── pc3
├── pc3.startup
├── router1
├── router1.startup
├── router2
├── router2.startup
└── shared

6 directories, 6 files
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ cat lab.conf
pc1[0]=A
pc2[0]=B
pc3[0]=C

router1[0]=A
router1[1]=B

router2[0]=A
router2[1]=C
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$
```



UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox

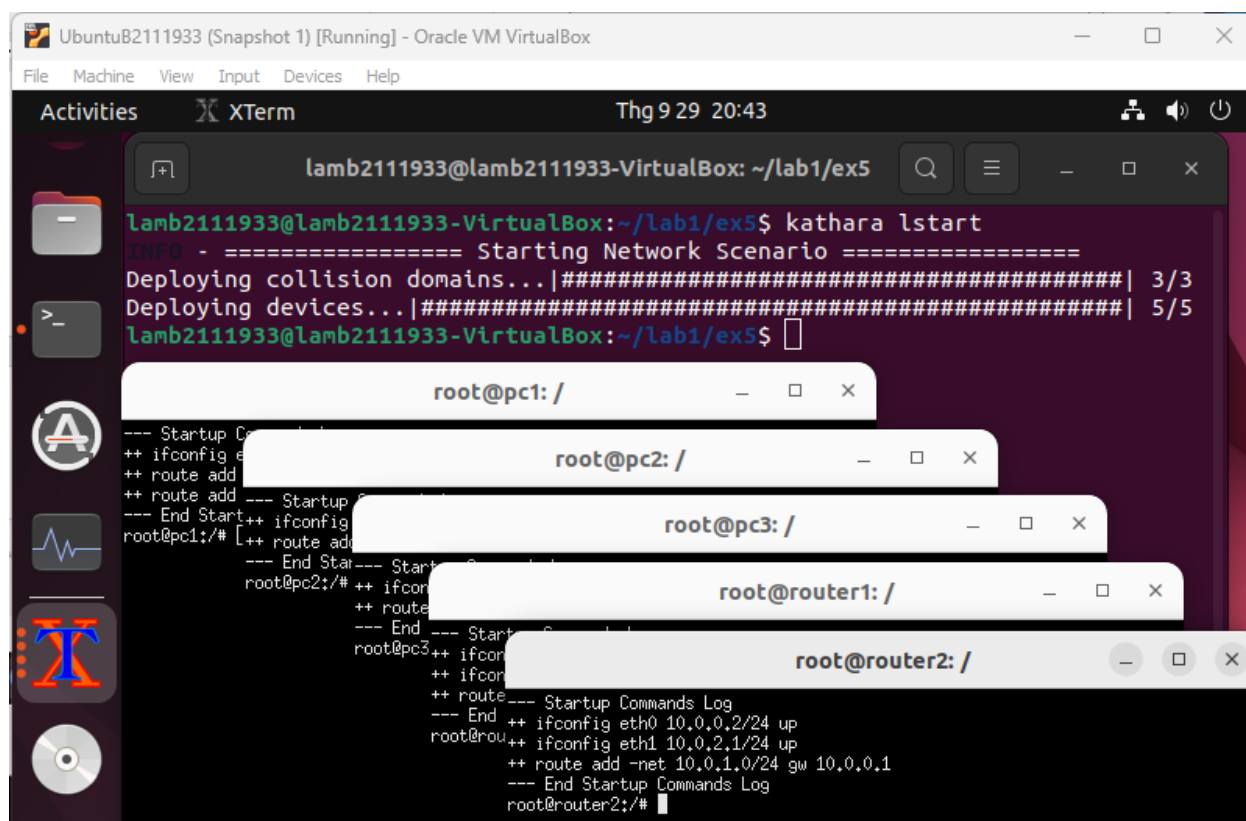
File Machine View Input Devices Help

Activities Terminal Thg 9 29 20:36

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex5

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ cat router1.startup
ifconfig eth0 10.0.0.1/24 up
ifconfig eth1 10.0.1.1/24 up
route add -net 10.0.2.0/24 gw 10.0.0.2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ cat router2.startup
ifconfig eth0 10.0.0.2/24 up
ifconfig eth1 10.0.2.1/24 up
route add -net 10.0.1.0/24 gw 10.0.0.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ cat pc1.startup
ifconfig eth0 10.0.0.101/24 up
route add -net 10.0.1.0/24 gw 10.0.0.1
route add -net 10.0.2.0/24 gw 10.0.0.2
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ cat pc2.startup
ifconfig eth0 10.0.1.101/24 up
route add default gw 10.0.1.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ cat pc3.startup
ifconfig eth0 10.0.2.101/24 up
route add default gw 10.0.2.1
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$
```

Start the lab:



UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities XTerm Thg 9 29 20:43

lamb2111933@lamb2111933-VirtualBox: ~/lab1/ex5

```
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$ kathara lstart
INFO - ===== Starting Network Scenario =====
Deploying collision domains...|#####| 3/3
Deploying devices...|#####| 5/5
lamb2111933@lamb2111933-VirtualBox:~/lab1/ex5$
```

root@pc1: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.101/24 up
++ route add -net 10.0.1.0/24 gw 10.0.0.1
++ route add -net 10.0.2.0/24 gw 10.0.0.2
--- End Startup Commands Log
root@pc1:/#
```

root@pc2: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.1.101/24 up
++ route add default gw 10.0.1.1
--- End Startup Commands Log
root@pc2:/#
```

root@pc3: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.2.101/24 up
++ route add default gw 10.0.2.1
--- End Startup Commands Log
root@pc3:/#
```

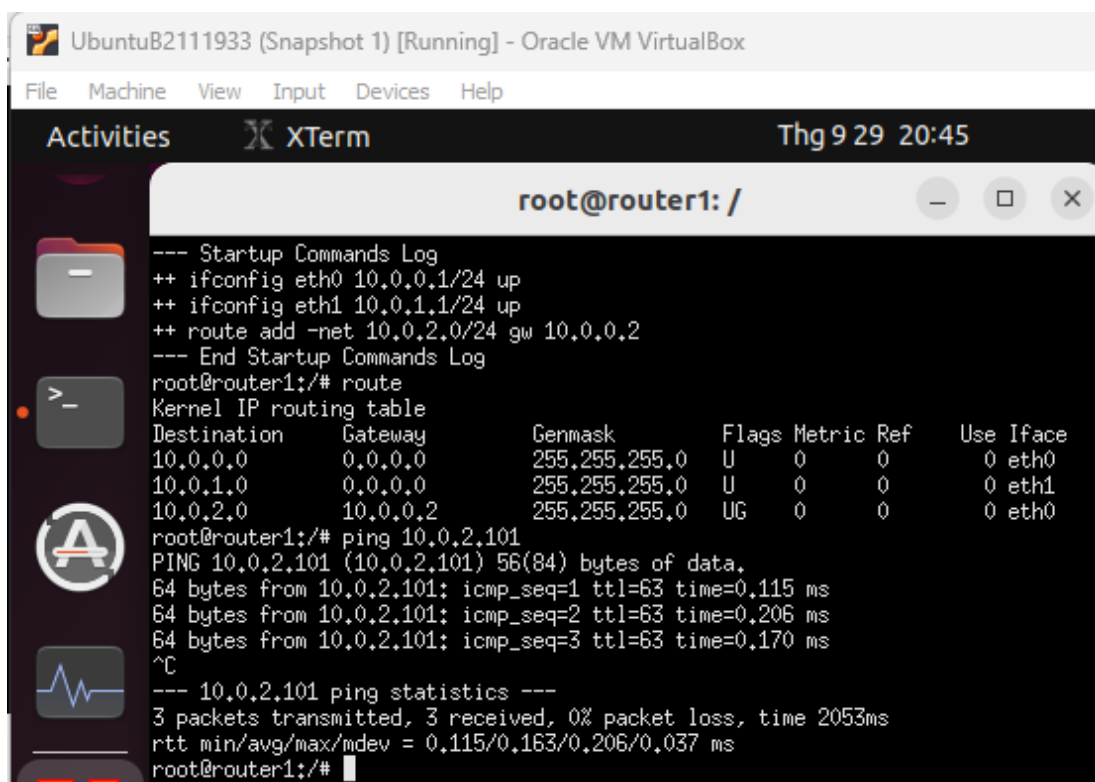
root@router1: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.1/24 up
++ ifconfig eth1 10.0.1.1/24 up
++ route add -net 10.0.2.0/24 gw 10.0.0.2
--- End Startup Commands Log
root@router1:/#
```

root@router2: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.2/24 up
++ ifconfig eth1 10.0.2.1/24 up
++ route add -net 10.0.1.0/24 gw 10.0.0.1
--- End Startup Commands Log
root@router2:/#
```


Check the result:



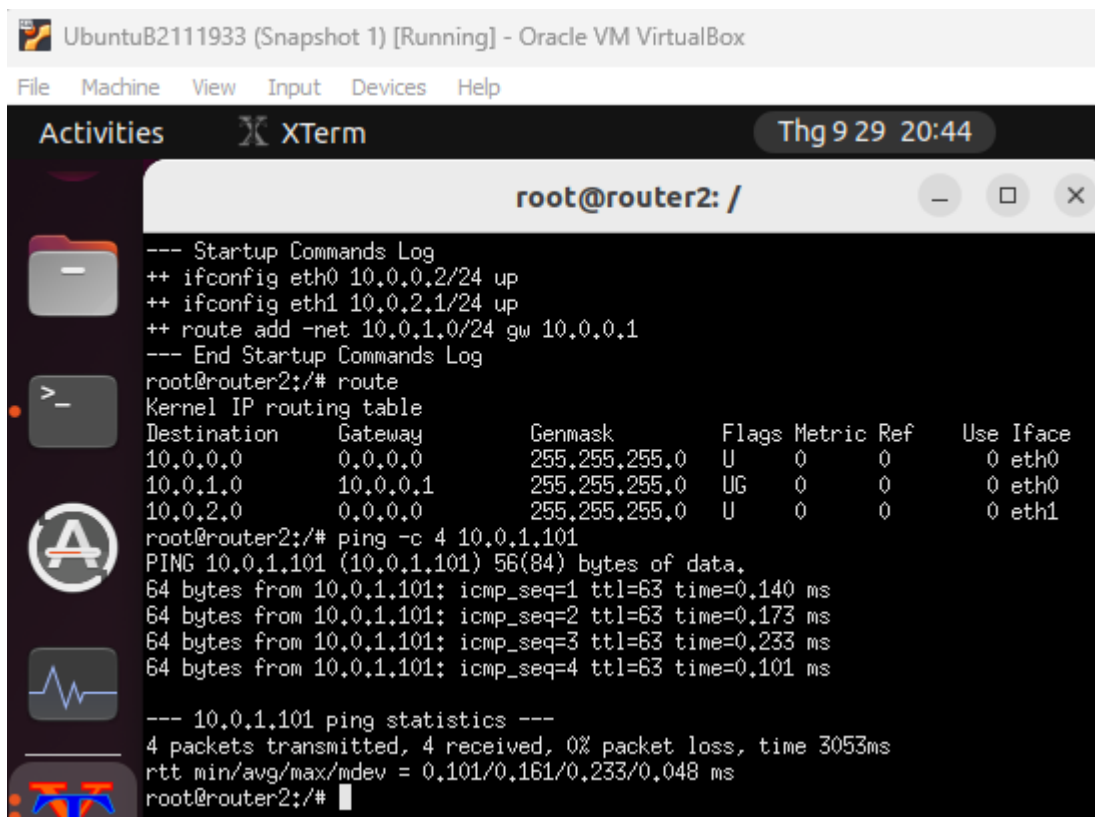
UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities XTerm Thg 9 29 20:45

root@router1: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.1/24 up
++ ifconfig eth1 10.0.1.1/24 up
++ route add -net 10.0.2.0/24 gw 10.0.0.2
--- End Startup Commands Log
root@router1:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
10.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
10.0.2.0 10.0.0.2 255.255.255.0 UG 0 0 0 eth0
root@router1:~# ping 10.0.2.101
PING 10.0.2.101 (10.0.2.101) 56(84) bytes of data.
64 bytes from 10.0.2.101: icmp_seq=1 ttl=63 time=0.115 ms
64 bytes from 10.0.2.101: icmp_seq=2 ttl=63 time=0.206 ms
64 bytes from 10.0.2.101: icmp_seq=3 ttl=63 time=0.170 ms
^C
--- 10.0.2.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.115/0.163/0.206/0.037 ms
root@router1:~#
```



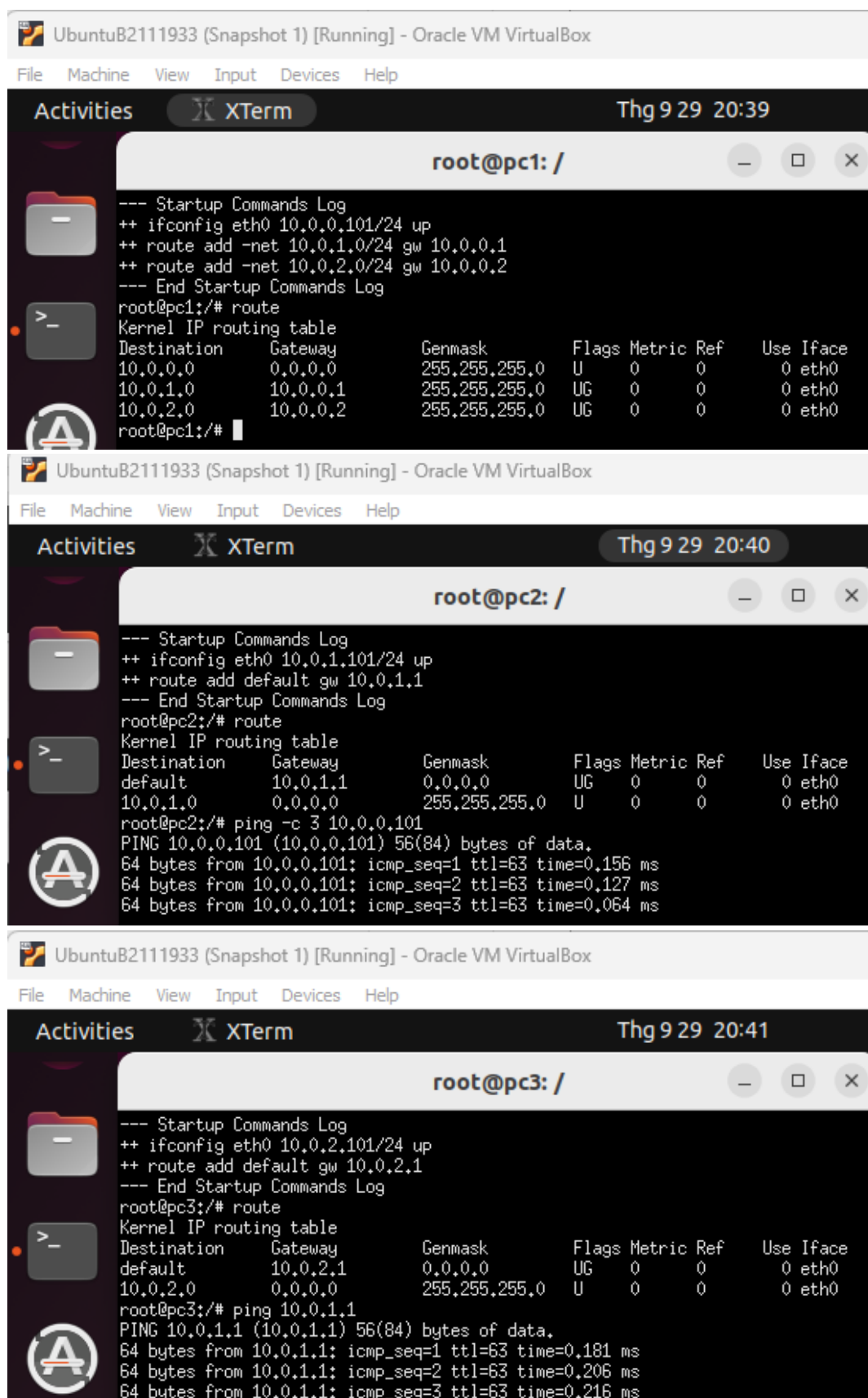
UbuntuB2111933 (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities XTerm Thg 9 29 20:44

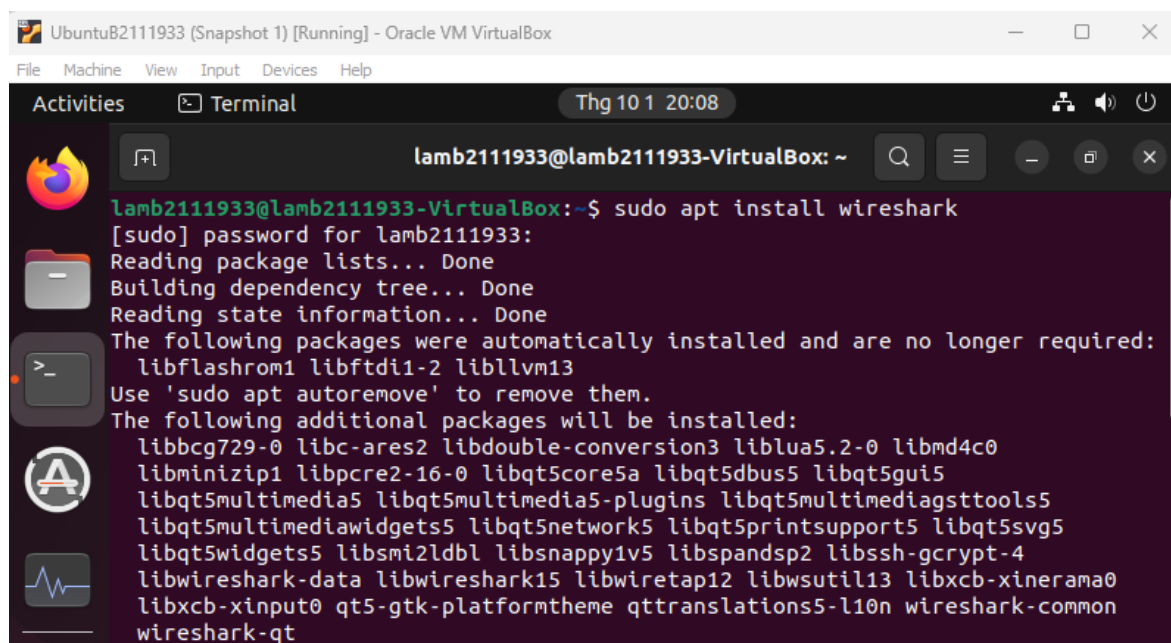
root@router2: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.2/24 up
++ ifconfig eth1 10.0.2.1/24 up
++ route add -net 10.0.1.0/24 gw 10.0.0.1
--- End Startup Commands Log
root@router2:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
10.0.1.0 10.0.0.1 255.255.255.0 UG 0 0 0 eth0
10.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
root@router2:~# ping -c 4 10.0.1.101
PING 10.0.1.101 (10.0.1.101) 56(84) bytes of data.
64 bytes from 10.0.1.101: icmp_seq=1 ttl=63 time=0.140 ms
64 bytes from 10.0.1.101: icmp_seq=2 ttl=63 time=0.173 ms
64 bytes from 10.0.1.101: icmp_seq=3 ttl=63 time=0.233 ms
64 bytes from 10.0.1.101: icmp_seq=4 ttl=63 time=0.101 ms
--- 10.0.1.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3053ms
rtt min/avg/max/mdev = 0.101/0.161/0.233/0.048 ms
root@router2:~#
```



Use Wireshark:

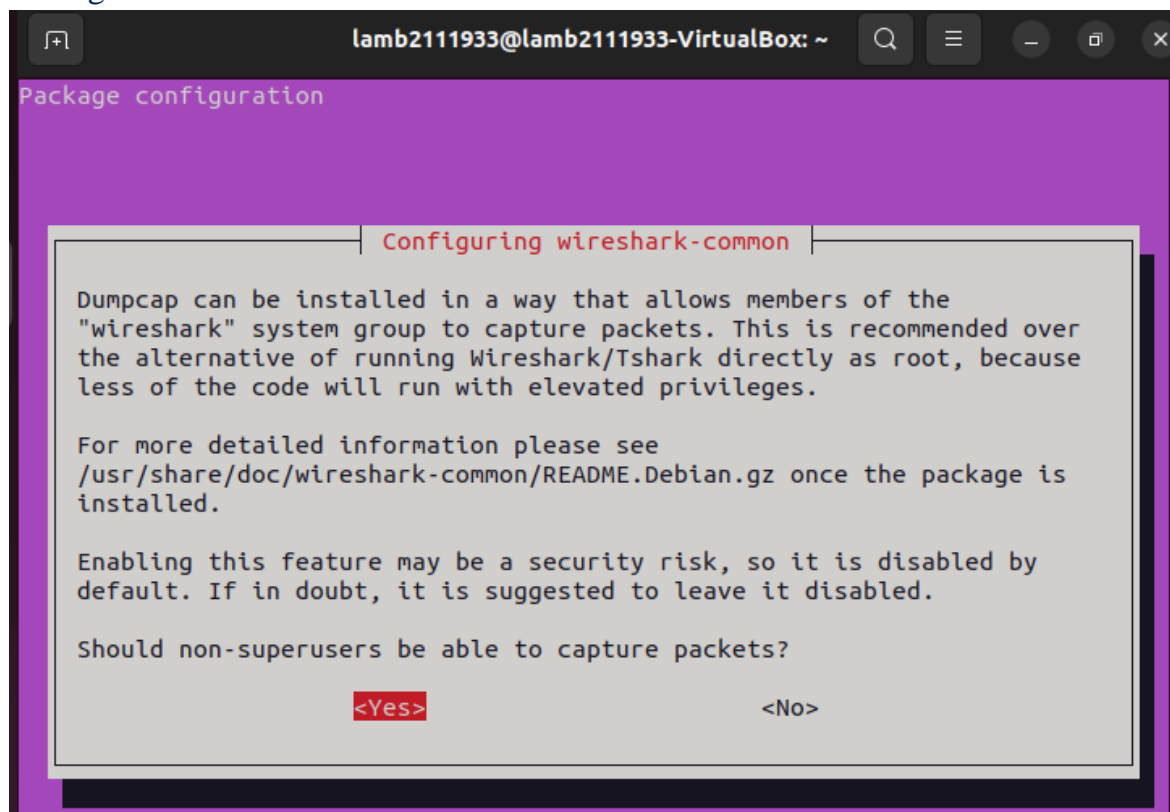
Install Wireshark



The screenshot shows a terminal window titled "UbuntuB2111933 [Snapshot 1] [Running] - Oracle VM VirtualBox". The terminal output shows the command `sudo apt install wireshark` being executed. The system prompts for a password, then lists the packages to be installed, including `libflashrom1`, `libftdi1-2`, `libllvm13`, and a long list of additional packages like `libbcb729-0`, `libc-ares2`, `libdouble-conversion3`, `liblua5.2-0`, `libmd4c0`, `libminizip1`, `libpcrc2-16-0`, `libqt5core5a`, `libqt5dbus5`, `libqt5gui5`, `libqt5multimedia5`, `libqt5multimedia5-plugins`, `libqt5multimediagsttools5`, `libqt5multimediawidgets5`, `libqt5network5`, `libqt5sprintsupport5`, `libqt5svg5`, `libqt5widgets5`, `libsmi2ldbl`, `libsnappy1v5`, `libspandsp2`, `libssh-gcrypt-4`, `libwireshark-data`, `libwireshark15`, `libwiretap12`, `libwsutil13`, `libxcb-xinerama0`, `libxcb-xinput0`, `qt5-gtk-platformtheme`, `qttranslations5-l10n`, `wireshark-common`, and `wireshark-qt`.

```
lamb2111933@lamb2111933-VirtualBox:~$ sudo apt install wireshark
[sudo] password for lamb2111933:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libbcb729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0
  libminizip1 libpcrc2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
  libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5
  libqt5multimediawidgets5 libqt5network5 libqt5sprintsupport5 libqt5svg5
  libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4
  libwireshark-data libwireshark15 libwiretap12 libwsutil13 libxcb-xinerama0
  libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n wireshark-common
  wireshark-qt
```

Configure Wireshark



The screenshot shows the "Configuring wireshark-common" dialog box. It contains the following text:

Dumpcap can be installed in a way that allows members of the "wireshark" system group to capture packets. This is recommended over the alternative of running Wireshark/Tshark directly as root, because less of the code will run with elevated privileges.

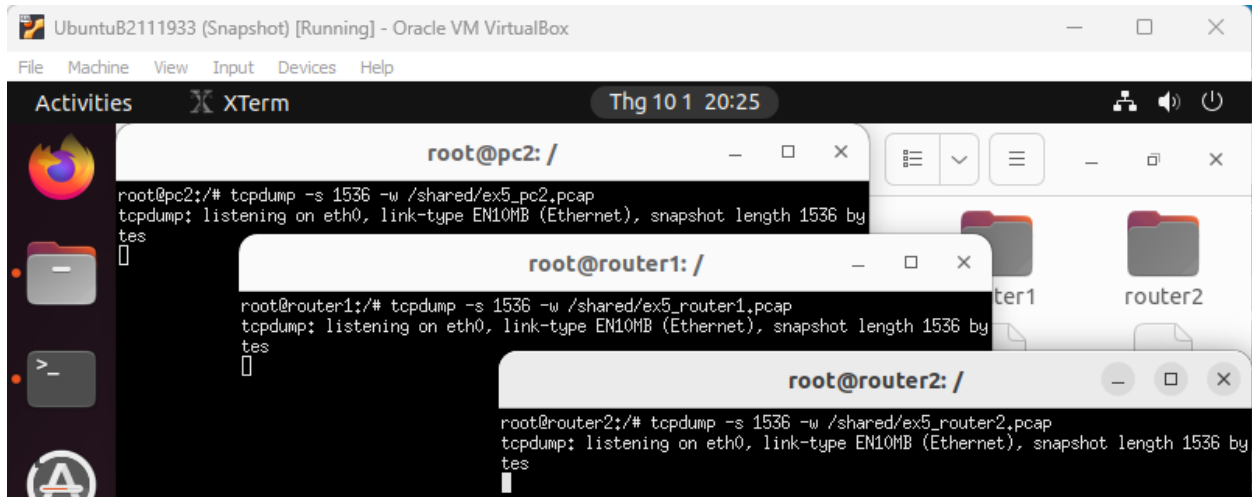
For more detailed information please see `/usr/share/doc/wireshark-common/README.Debian.gz` once the package is installed.

Enabling this feature may be a security risk, so it is disabled by default. If in doubt, it is suggested to leave it disabled.

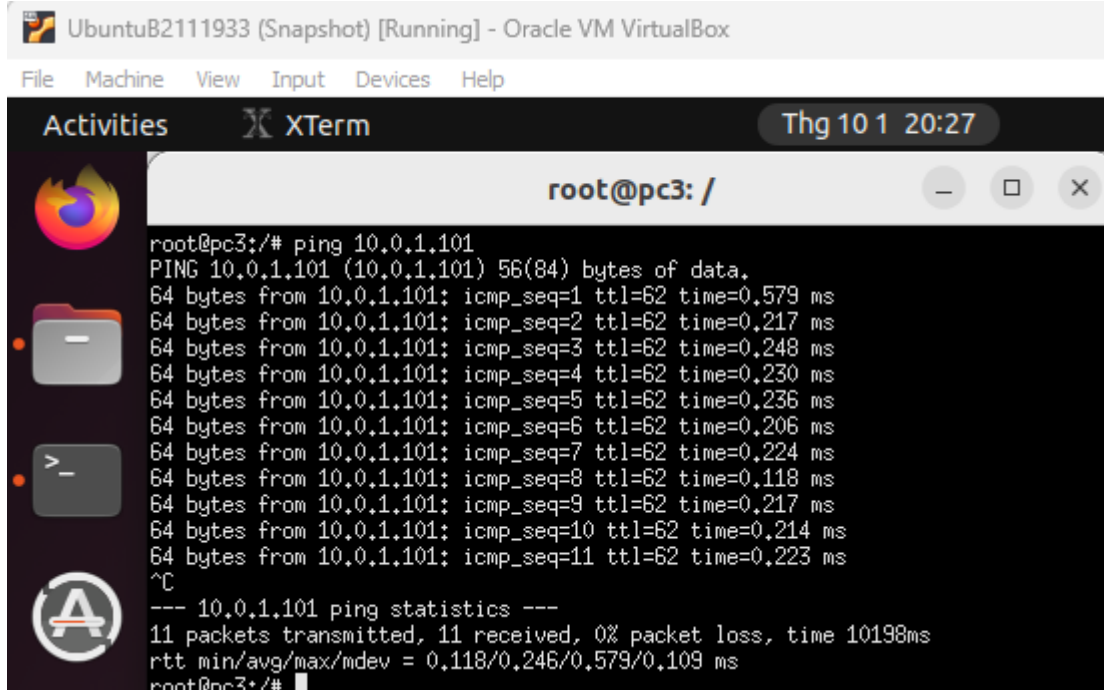
Should non-superusers be able to capture packets?

The dialog has two buttons: "<Yes>" and "<No>".

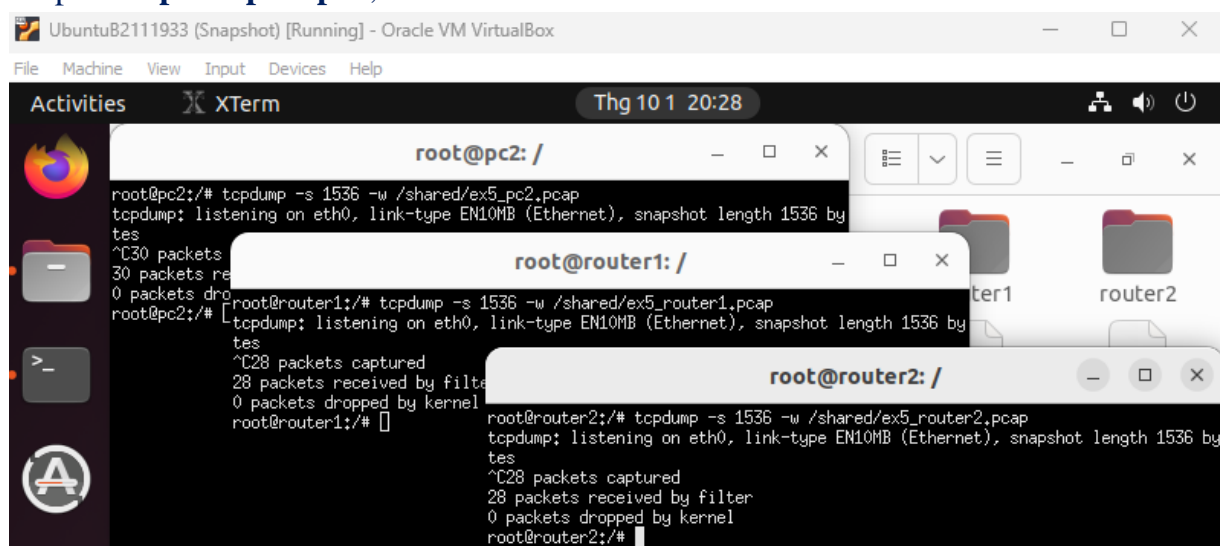
All packets are save in **.pcap** files which are in the **/shared** folder



On **pc3**, send packets to **pc2** using the command **ping 10.0.1.101**



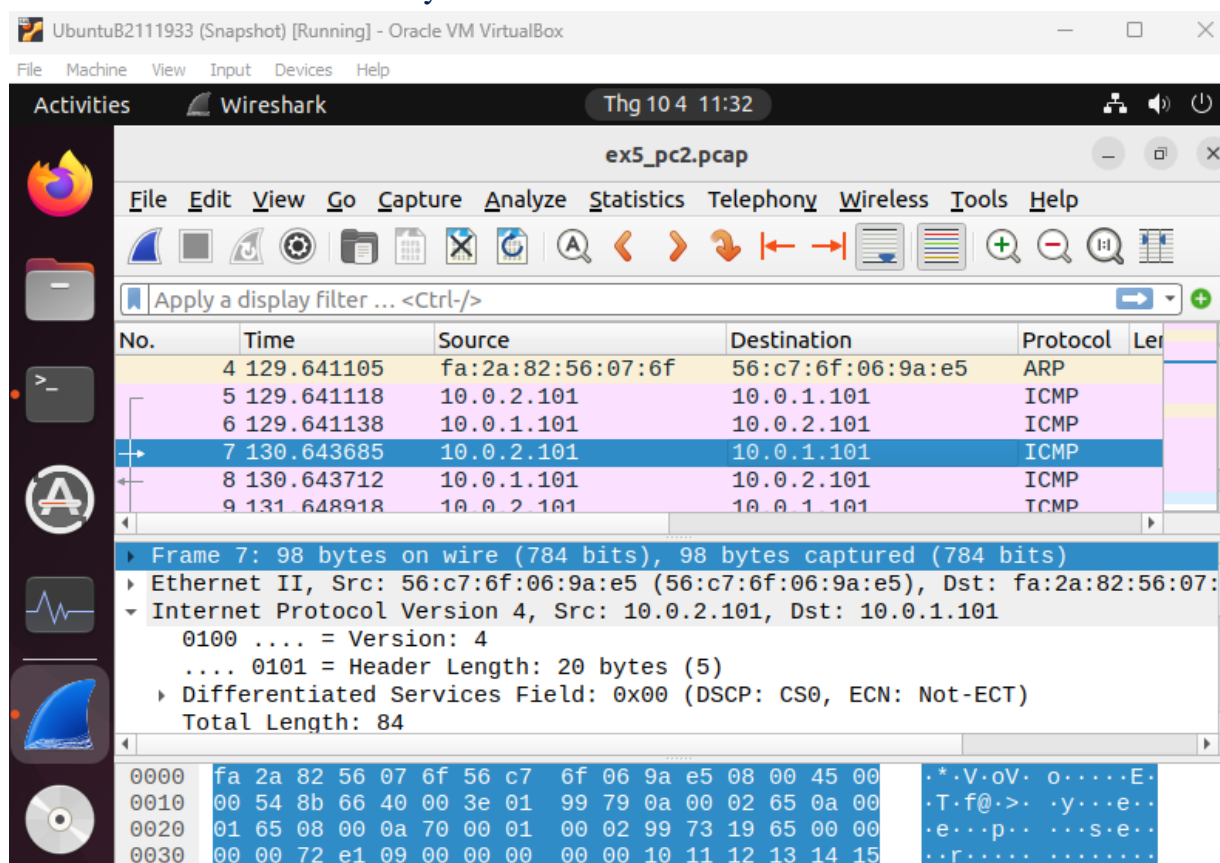
Stop the **tcpdump** on **pc2**, **router1** and **router2**



On the Ubuntu, open **Ex5_pc2.pcap** using **Wireshark**, select the frame #7 and answer the following questions:

- Size of frame in bytes?

Answer: Size of frame is 98 bytes



- Select Header Internet Control Message Protocol → which protocol is using? On which layer of the OSI model does this protocol operate? What is the content of the message? How long is this message in bytes?

Answer:

- The protocol is **ICMP**
- This protocol operates on **network layer**
- The content of the message is below
- Size of this message is **48 bytes**

The screenshot shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for packet capture and analysis. The main display area shows a list of captured packets in a table with columns: No., Time, Source, Destination, Protocol, and Length. Packet 7 is selected, showing an ICMP Echo (ping) request from 10.0.2.101 to 10.0.1.101. The packet details pane on the right shows the expanded ICMP header with fields: Type: 8 (Echo (ping) request), Code: 0, Checksum: 0x0a70 [correct], Identifier (BE): 1 (0x0001), Identifier (LE): 256 (0x0100), Sequence Number (BE): 2 (0x0002), Sequence Number (LE): 512 (0x0200), and a timestamp. The packet data pane at the bottom shows the raw data in hexadecimal and ASCII, with the first 48 bytes highlighted in blue.

No.	Time	Source	Destination	Protocol	Length
1	0.000000	fe80::8881:5fff:fed...	ff02::2	ICMPv6	
2	61.440196	fe80::54b4:13ff:fec...	ff02::2	ICMPv6	
3	129.641093	56:c7:6f:06:9a:e5	Broadcast	ARP	
4	129.641105	fa:2a:82:56:07:6f	56:c7:6f:06:9a:e5	ARP	
5	129.641118	10.0.2.101	10.0.1.101	ICMP	
6	129.641138	10.0.1.101	10.0.2.101	ICMP	
7	130.643685	10.0.2.101	10.0.1.101	ICMP	

Internet Control Message Protocol

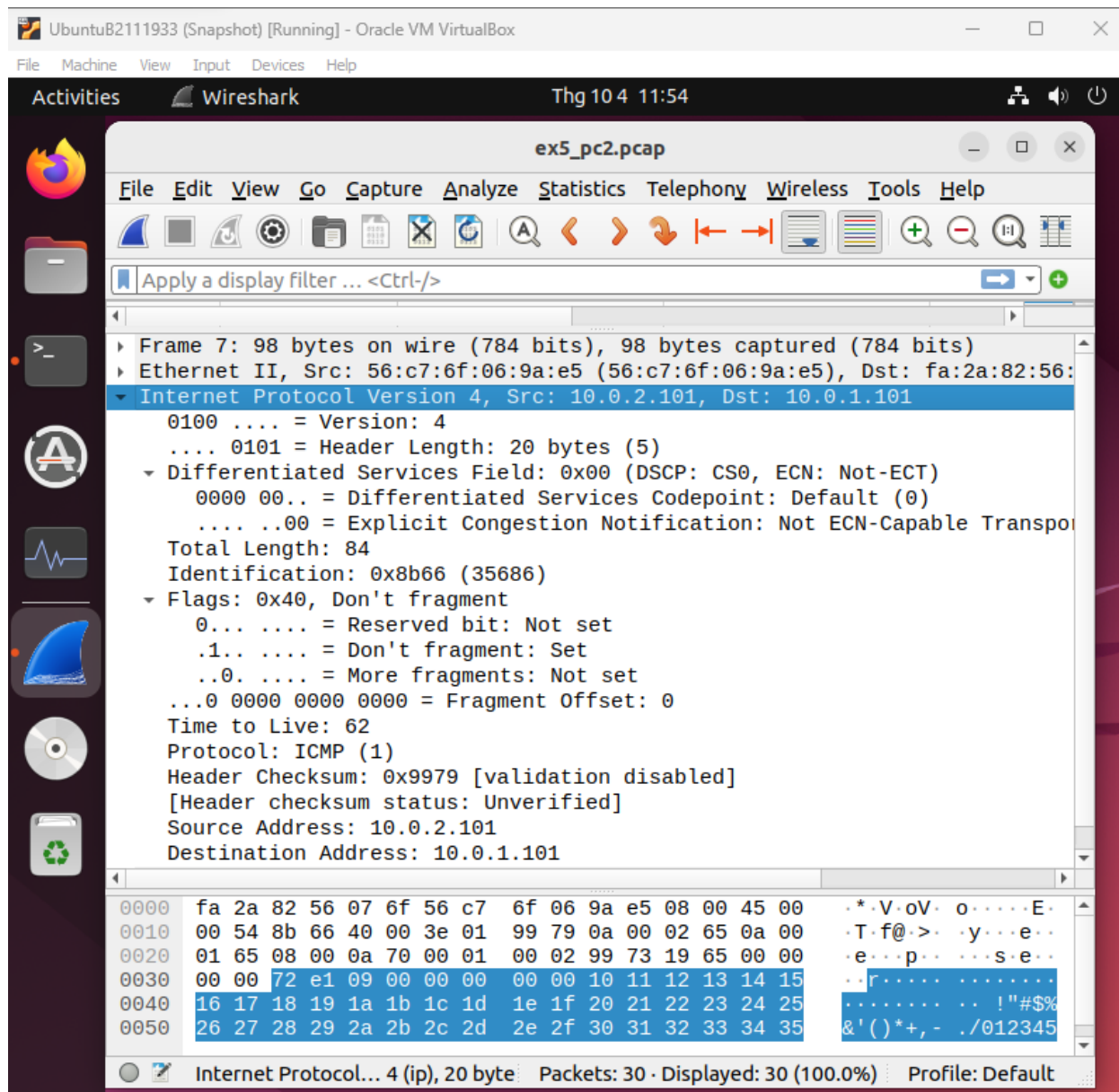
- Type: 8 (Echo (ping) request)
- Code: 0
- Checksum: 0x0a70 [correct]
- [Checksum Status: Good]
- Identifier (BE): 1 (0x0001)
- Identifier (LE): 256 (0x0100)
- Sequence Number (BE): 2 (0x0002)
- Sequence Number (LE): 512 (0x0200)
- [Response frame: 8]
- Timestamp from icmp data: Oct 1, 2023 20:26:49.000000000 +07
- [Timestamp from icmp data (relative): 0.647688000 seconds]

Data (48 bytes)

Data: 72e1090000000000101112131415161718191a1b1c1d1e1f202122232425262728

0000 fa 2a 82 56 07 6f 56 c7 6f 06 9a e5 08 00 45 00 .*.V.oV.o....E.
 0010 00 54 8b 66 40 00 3e 01 99 79 0a 00 02 65 0a 00 .T.f@.>.y...e..
 0020 01 65 08 00 0a 70 00 01 00 02 99 73 19 65 00 00 .e...p...s.e..
 0030 00 00 72 e1 09 00 00 00 00 00 10 11 12 13 14 15 .r.....
 0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25!"\$%&'()*+,-./012345
 0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35
 0060 36 37 67

Select Header Internet Protocol Version 4:



→ what are the IP addresses of the source and destination hosts?

Answer: IP addresses: source: **10.0.2.101** destination: **10.0.1.101**

→ What is the length of the IP packet header? What fields does the Header include?

How long is each field (Bytes)

Answer:

- The length of the IP packet header: 20 bytes
- Fields of the Header:
 - Version & Header Length (1 byte)
 - Differentiated Services Codepoint & Explicit Congestion Notification (1 byte)

- Total Length (2 bytes)
- Identification (2 bytes)
- Flags (2 bytes)
- Time To Live (1 byte)
- Protocol (1 byte)
- Header Checksum (2 bytes)
- Source Address (4 bytes)
- Destination Address (4 bytes)

→ What is the length of the Total Length field (Bytes).

Answer: the length of the Total Length field is **2 bytes**

Select Header Ethernet II:

The screenshot shows the Wireshark interface with the following details:

- Packet List:**

No.	Time	Source	Destination	Protocol
7	130.643685	10.0.2.101	10.0.1.101	ICMP
8	130.643712	10.0.1.101	10.0.2.101	ICMP
- Packet Details (Packet 8):**
 - Frame 7: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
 - Ethernet II, Src: 56:c7:6f:06:9a:e5 (56:c7:6f:06:9a:e5), Dst: fa:2a:82:56:07:6f (fa:2a:82:56:07:6f)
 - Destination: fa:2a:82:56:07:6f (fa:2a:82:56:07:6f)
 - Address: fa:2a:82:56:07:6f (fa:2a:82:56:07:6f)
 -1. = LG bit: Locally administered address (true)
 -0. = IG bit: Individual address (unicast)
 - Source: 56:c7:6f:06:9a:e5 (56:c7:6f:06:9a:e5)
 - Address: 56:c7:6f:06:9a:e5 (56:c7:6f:06:9a:e5)
 -1. = LG bit: Locally administered address (true)
 -0. = IG bit: Individual address (unicast)
 - Type: IPv4 (0x0800)
 - Internet Protocol Version 4, Src: 10.0.2.101, Dst: 10.0.1.101
 - Internet Control Message Protocol
- Packet Bytes:**

Offset	Hex	ASCII
0000	fa 2a 82 56 07 6f 56 c7 6f 06 9a e5 08 00 45 00	..V.oV. o....E.
0010	00 54 8b 66 40 00 3e 01 99 79 0a 00 02 65 0a 00	.T.f@.> .y...e..
0020	01 65 08 00 0a 70 00 01 00 02 99 73 19 65 00 00	.e...p...s.e...
0030	00 00 72 e1 09 00 00 00 00 00 10 11 12 13 14 15	..r.....
0040	16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !"#\$%

→ What are the MAC addresses of the source and the destination hosts?

Answer: MAC addresses: - source: 56:c7:6f:06:9a:e5
- destination: fa:2a:82:56:07:6f

→ What is the Type value?

Answer: Type: IPv4 (0x0800)